ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF TECHNOLOGY

FACE RECOGNITION
USING ARTIFICIAL NEURAL NETWORK

BY

Sentayehu Endeshaw Wolde

August, 2006

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF TECHNOLOGY

FACE RECOGNITION

USING ARTIFICIAL NEURAL NETWORK

A thesis submitted to the School of Graduate Studies of Addis Ababa University in partial fulfillment for the Degree of Master of Science in Computer Engineering

By

Sentayehu Endeshaw Wolde

Advisor

Dr. Kumudha Raimond

Addis Ababa, Ethiopia

August, 2006

ADDIS ABABA UNIVERSITY

SCHOOL OF GRADUATE STUDIES

FACULTY OF TECHNOLOGY

FACE RECOGNITION

USING ARTIFICIAL NEURAL NETWORK

BY

Sentayehu Endeshaw Wolde

Approval by Board of Examiners

Dr. Getachew Biru Chairman, Department of Electrical and Computer Engineering	_____ Signature
Dr. Kumudha Raimond Advisor	_____ Signature
_____ External Examiner	_____ Signature
_____ Internal Examiner	_____ Signature

Abstract

In recent years, an explosion in research on pattern recognition systems using neural network methods has been observed. Face Recognition (FR) is a specialized pattern recognition task for several applications such as security: access to restricted areas, banking: identity verification and recognition of wanted people at airports.

This thesis will explain what is involved in FR task and outline a complete Face Recognition System (FRS) based on Artificial Neural Network (ANN). In this work, two FRS are developed. The first model uses Principal Component Analysis (PCA) for feature extraction from the face images and ANN for the classification purpose. In the second model, combination of Gabor Filter (GF) and PCA are used for feature extraction and ANN for the classification.

In the first approach, the face images are projected into subspace called eigenspace, consisting of the eigenvectors from the covariance matrix of the face images. The projection of an image into eigenspace will transform the image into a representation of a lower dimension which aims to hold the most important features of the face. These feature vectors are classified into training, validation and testing set. The training and validation set are used during the training of ANN. The testing set is used to evaluate the recognition performance of the model.

In the second approach, Gabor feature vectors are derived from a set of downsampled Gabor wavelet representations of face images, then the dimensionality of the vectors is reduced by means of Principal Component Analysis (PCA), and finally ANN is used for classification. The Gabor filtered face images exhibit strong characteristics of spatial locality, scale, and orientation selectivity. These images can, thus, produce salient local features that are most suitable for FR.

Experimentation is carried out on FRS by using Olivetti Research Laboratory (ORL) datasets, the images of which vary in illumination, expression, pose and scale. The result

shows the feasibility of the methodology followed in this thesis work. Model 1 achieves a recognition rate of 76.6% whereas model 2 achieves 88.3% of correct classification and performed very efficiently when subjected to new unseen images with a false rejection rate of 0% during testing. The high recognition rate of model 2 shows the efficiency of GF in feature extraction.

Key words—Face recognition, biometrics, artificial neural network, Gabor filter and principal component analysis.

Dedicated to my mother

ACKNOWLEDGEMENTS

First of all, I would like to thank God for giving me courage and strength to finish this work. Also, I would like to thank my advisor, Dr. Kumudha Raimond, for her continued guidance and support during the course of this thesis. Dr. Kumudha's vision and her continued constructive criticisms have helped me accomplish my goals. Without her this work would not have been possible. I would really like to thank her for her patience and time to guide me at every step of this work.

I would also like to thank my relatives for their moral support. I have no words to describe the support offered to me by my friends especially Fitsum Tilahun.

Words cannot express my deepest gratitude to my parents, who motivated me to take up this challenge and helped me complete it with their love and support.

I am deeply thankful to my friends at Addis Ababa University ICT Development office for their support in this study. I would also like to thank the Olivetti Research Laboratory for the availability of their face database which is used for this thesis.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Acronyms

FR	Face Recognition
FRS	Face Recognition System
ANN	Artificial Neural Network
ORL	Olivetti Research Laboratory
MLP	Multi Layer Perceptron
FERET	FacE REcognition Technology
PCA	Principal Component Analysis
GF	Gabor Filter
PE	Processing Element
BPN	Back Propagation Network
ALC	Adaptive Linear Combiner
HVS	Human Visual System
FFT	Fast Fourier Transform
HMM	Hidden Markov Model
FAR	False Acceptance Rate
FRR	False Rejection Rate
ERR	Equal Error Rate
ROC	Receiver Operating Characteristic

CHAPTER 1 INTRODUCTION

Machine recognition of faces is emerging as an active research area spanning several disciplines such as image processing, pattern recognition, computer vision and ANN. FR technology has numerous commercial and law enforcement applications. These applications range from static matching of controlled format photographs such as passports, credit cards, photo ID's, driver's license, and mug shots to real time matching of surveillance video images [1].

Humans seem to recognize faces in cluttered scenes with relative ease, also having the ability to identify distorted images, coarsely quantized images and faces with occluded details. Machine recognition is a much more daunting task. Understanding the human mechanisms employed to recognize faces constitute a challenge for psychologists and neural scientists. In addition to the cognitive aspects, understanding FR is important, since the same underlying mechanisms could be used to build a system for the automatic identification of faces by machine.

A formal method of classifying faces was first proposed by Francis Galton in 1888 [2, 3]. During 1980's, work on FR remained largely dormant. However, during 1990's, the research interest in FR has grown significantly as a result of the following facts:

1. The increase in emphasis on civilian/commercial research projects,
2. The emergence of ANN classifiers with emphasis on real time computation and adaptation,
3. The availability of real time hardware,
4. The increasing need for surveillance related applications due to drug-trafficking, terrorist activities, etc.

Still most of the access control methods, with all their legitimate applications in an expanding society, have a bothersome drawback. Except for face and voice recognition, these methods require the user to remember a password, to enter a PIN code, to carry a batch, or, in general, require a human action in the course of identification or authentication. In addition, the corresponding means (keys, batches,

passwords, PIN codes) are prone to being lost or forgotten, whereas fingerprints and retina scans suffer from low user acceptance. Modern FR has reached an identification rate of greater than 90% with well-controlled pose and illumination conditions. While this is a high rate for FR, it is not comparable to methods using keys, passwords or batches [43].

1.1 Why face recognition?

The modern information age confronts humanity with various challenges that did not exist to the same extent in earlier times. Two such challenges are the organization of society and security. Of the various different methods used to tackle these challenges and enforce the resulting measures, identification and authentication have a special status in modern society. In this context, *identification* means the determination of the identity of an individual, whereas *authentication* means the confirmation of this identity.

In earlier times and societies, where mobility was low and business was done on the basis of personal acquaintance, personal identification needs for traveling, legal documents, or bank affairs were satisfied either by rare identification papers or, mostly, by seals or signatures. These means, which were often reserved for a certain social class, allowed a sufficiently safe and efficient organization and handling of daily affairs and business.

This has changed with this century's industrial and technological development and the exponential growth of the world population. The newfound wealth and technology of this evolution has allowed ever increasing human mobility in all its facets. This mobility in turn caused an increasing demand for enhanced ways of sharing information and transferring data, and soon enough, through complex communication structures, information gained its own mobility - a mobility that even outgrew its human counterpart.

In this context of increased mobility and world population, security and organization have become an important social issue. As mobility applies to both humans and information, security also applies to both individuals and their valuables, and the integrity of data under external influence.

Within this environment of increased importance of security and organization, identification and authentication methods have developed into a key technology in various areas: entrance control in buildings; access control for computers in general or for automatic teller machines in particular; day-to-day affairs like withdrawing money from a bank account or dealing with the post office; or in the prominent field of criminal investigation. These few examples illustrate the necessity of identification and authentication for the functioning of modern society.

Different eras and cultures met these needs in different ways. While identity cards or PIN codes are the main method in bank affairs in most countries, the Chinese, for instance, still use seals as their primary authentication means in daily life. Other means include human recognition (e.g. a concierge in a building), passports, keys, batches, or passwords. More sophisticated applications use fingerprints or retina scans, basic voice recognition, or a combination of the aforementioned techniques.

All these methods are under constant development, while new ones are being investigated. At the same time, identity cards are made less forgeable, keys and batches work in smarter environments, and fingerprint or retina scan methods reached an increased degree of automation.

Still, most of these methods, with all their legitimate applications in an expanding society, have a bothersome drawback. Except for face and voice recognition, these methods require the user to remember a password, to enter a PIN code, to carry a batch, or, in general, require a human action in the course of identification or authentication. In addition, the corresponding means (keys, batches, passwords, PIN codes) are prone to be lost or forgotten, whereas fingerprints and retina scans suffer from low user acceptance.

Modem FR has reached an identification rate of greater than 90% for large databases of images with well-controlled pose and illumination conditions [15]. While this is a high rate for FR, it is by no means comparable to methods using keys, batches or passwords, nor can it bear direct comparison with the recognition abilities of human beings. Still, FR as an identification or authentication means could be successfully employed in many of

the aforementioned tasks to support other techniques, or even replace them in the case of lower security requirements.

Apart from these more classical FR applications, in the modem information age, where computers are common tools for daily work processes, new authentication tasks keep appearing. In bank offices, for example, or in other institutions where security demand is higher than in ordinary offices, security breeches result from logged-in but temporarily unattended computers. Such security needs could be met nowadays, as many computers are already equipped with video cameras (or cameras could be easily installed). In this case, special FR software could constantly observe and match the face in front of the camera against the face of the user who claims to be logged in.

In the future, this kind of constant observation could be employed in a gesture and facial expression recognition environment, thus turning "personal" computers into "personalized" computers able to interact with the user on a level higher than mouse clicks and key strokes. Together with software that could "understand" gestures and facial expressions, such recognition would make possible intelligent man-machine interfaces and, in the future, intelligent interaction with robots. [15]

1.2 Problem Definition

A general statement of the problem can be formulated as follows: given still or video images of a scene, identify one or more persons in the scene using a stored database of faces.

The environment surrounding a FR application can cover a wide spectrum from a well-controlled environment to an uncontrolled one. In a controlled environment, frontal and profile photographs are taken with uniform background and identical poses among the participants. These face images are commonly called mug *shots*. Each mug shot can be manually or automatically cropped to extract a normalized subpart called a canonical face image. In a canonical face image, the size and position of the face are normalized approximately to the predefined values and background region is minimal.

FR, a task that is done by humans as daily activity, comes from virtually uncontrolled environment. Systems, which automatically recognize faces from uncontrolled environment, must detect faces in images. Face detection task is to report the location, and typically also the size, of all the faces from a given image and this is completely a different problem with respect to FR.

FR is a difficult problem due to the general similar shape of faces combined with the numerous variations between images of the same face. Recognition of faces from an uncontrolled environment is a very complex task: lighting condition may vary tremendously; facial expressions also vary from time to time; face may appear at different orientations and a face can be partially occluded. Further, depending on the application, handling facial features over time (aging) may also be required. Although existing methods perform well under constrained conditions, the problems with the illumination changes, out of plane rotations and occlusions still remain unsolved. Since the techniques used in the best FRS may depend on the application of the system, one can identify at least two broad categories of FRS [4]:

1. Finding a person within large database of faces (e.g. in a police database). (Often only one image is available per person. It is usually not necessary for recognition to be done in real time.)

2. Identifying particular people in real time (e.g. location tracking system). (Multiple images per person are often available for training and real time recognition is required.)

In this thesis, the primarily interest is in the first case. The aim is to provide the correct label (e.g. name of the person) associated with the new faces during the recognition phase in case of occlusions and illumination changes. The objective of this work is to report on the performance of two different feature extraction techniques and on the issues in building the ANN model for FRS. A significant part of this report shows the methodology of how the architecture and parameters of the ANN system are chosen.

The system uses the standard face database, ORL face database for testing the performance of FRS. In the initial stage, two feature extraction techniques such as PCA and GF are used to extract the feature vectors. MLP, which is one of the popular ANN models, is used in the final stage for recognition.

1.3 Organization of the Thesis

The overall organization of this thesis is as follows. Chapter 2 discusses about biometrics, and its various types and comparisons between them, besides, deals with biometric systems and related issues. Chapter 3 deals with both the theoretical and practical issues of ANN. Chapter 4 introduces FR, Face Acquisition, Face Representation and Face Reasoning techniques that are used in this thesis. In Chapter 5 and 6, experimental results corresponding to feature extraction techniques and ANN model development are presented. In chapter 7, concluding remarks and recommendations are presented.

CHAPTER 2 BIOMETRICS

2.1 Introduction

Because biometrics can be used in a variety of applications, it is difficult to establish an all-encompassing definition. The older sense of biometrics (also known as biometry), from Encyclopedia Britannica XXVIII; 1902, was “The application of modern statistical methods to the measurements of biological (variable) objects” [16]. This however should not be confused with one of the newer definitions, which is stated as “The identification of an individual based on biological traits, such as fingerprints, iris patterns, and facial features” [16]. This newer sense gives a much more accurate explanation on what biometrics is all about nowadays; which is identification of individuals. Therefore, the most suitable and concise definition of biometrics is “the automatic recognition of a person using distinguishing traits” or a more expansive definition of it (biometrics) is “any automatically measurable, robust and distinctive physical (physiological) characteristics or personal trait (behavioral) characteristics that can be used to identify an individual or verify the claimed identity of an individual.” [13]

Measurable means that the characteristics or trait can be easily acquired by a sensor, and converted into a quantifiable, digital format. This measurability allows for matching to occur in a matter of seconds and makes it an automated process [17, 18].

The *robustness* of a biometric refers to the extent to which the characteristics or trait is subjected to significant changes over time. These changes can occur as a result of age, injury, illness, occupational use, or chemical exposure. A highly robust biometric does not change significantly over time while a less robust biometric will change [17]. For example, the iris, which changes very little over a person’s lifetime, is more robust than one’s voice [13, 14].

Distinctiveness is a measure of the variations or differences in the biometric patterns among the general population. The higher the degree of distinctiveness, the more the biometric characteristic is an identifier. A low degree of distinctiveness indicates a

biometric pattern found frequently in the general population. The iris and the retina have higher degree of distinctiveness than hand or finger geometry [14, 17].

2.2 History and development

Biometrics is defined as measurable physiological and/or behavioral characteristics. The history of biometrics includes the identification of people by unique body features, scars or a combination of other physiological criteria, such as height, eye color and complexion. Early uses of biometrics include the practice in ancient China, whereby babies were distinguished from each other through ink stamps of palm and footprints. In the early nineteenth century, criminology was the main driver of biometrics, when researchers studied the relationship between physical features and criminal tendencies. A method called anthropometrical signalment involved taking measurements of people's skull to identify criminals and catch frequent offenders. Although no definitive conclusions were reached on the link between cranial features and crime, this work did lead to the use of the most well known biometric as the international standard for identification.

Although at present biometrics has limited mainstream usage, biometrics has found a home in popular culture, specifically the movies. Movies have used biometrics in science fiction or adventure films, including such movies as Total Recall and Charlie's Angels. Examples of biometrics in movies include forged identities through high tech facemasks, voice disguise, forged hands or fingerprints, even false retinal images through the use of contact lenses. Whether the movies are prophetic in depicting how easily biometrics can be circumnavigated remains to be seen. Without question, as with all security measures, there will always be those who seek to evade detection.

Currently, biometric techniques are used mainly in security operations. For example, they are used in prison visitor system, state benefit payment system, border control, gold and diamond mines and bank vaults. Clearly these are the areas where security is an issue and fraud is a threat. Recent world events have led to an increased interest in security that will propel biometrics into mainstream use. Areas of use include

workstation and network access, Internet transactions, telephone transactions and in travel and tourism.

There are a number of different types of biometrics: Some are ages old; others are more recent and employ the latest technology. Technological advances will surely refine existing methods and lead to the development of new ones. The most well known biometric technologies include fingerprint, hand geometry, signature verification, voice verification, retina scanning, iris scanning and facial recognition.

2.3 A Comparison of Various Biometrics

As it is discussed above, any human physiological and/or behavioral characteristic can be used as a biometric identifier to recognize a person as long as it satisfies the following characteristics and requirements [12]:

- *Universality*, which means that each person should have the biometrics;
- *Distinctiveness*, which indicates that any two persons should be sufficiently different in terms of their biometric identity ;
- *Permanence*, which means that the biometrics should be sufficiently invariant (with respect to the matching criterion) over a period of time ;
- *Collectability*, which indicates that the biometrics can be measured quantitatively. However, in a practical biometric system, there are a number of other issues that should be considered, including:
- *Performance*, which refers to the achievable recognition accuracy, speed, robustness, the resource requirements to achieve the desired recognition accuracy and speed, as well as operational or environmental factors that affect the recognition accuracy and speed ;
- *Acceptability*, which indicates the extent to which people are willing to accept a particular biometric identifier in their daily lives ;
- *Circumvention*, which reflects how easy it is to fool the system by fraudulent methods

A practical biometric system should have acceptable recognition accuracy and speed with reasonable resource requirements, harmless to the users, accepted by the intended population, and sufficiently robust to various fraudulent methods [12, 13].

2.3.1 The Best Biometrics

A number of biometric identifiers are in use in various applications (Figure 2-1). Each biometric has its strengths and weaknesses and the choice typically depends on the application. No single biometric is expected to effectively meet the requirements of all the applications. If one specifically defines an application, it may be possible to describe the most accurate, easiest to use, easiest to deploy, or cheapest biometric for that particular application, but no one biometric technology or set of criteria is right for all situations. Hence, the match between a biometric method and an application is determined depending upon the characteristics of the application and the properties of the biometric method [12].

Therefore, when choosing a biometric for an application the following issues have to be addressed [12, 19]:

- Does the application need verification or identification? If an application requires an identification of a person from a large database, it needs a scalable and relatively more distinctive biometric method.
- What are the operational modes of the application? For example, whether the application is attended (semi-automatic) or unattended (fully automatic), whether the users are habituated (or willing to be habituated) to the given biometrics, whether the application is covert or overt, whether the subjects are cooperative or non-cooperative, and so on.
- What is the storage requirement of the application? For example, an application that performs the recognition at a remote server may require a small template size.
- How stringent are the performance requirements? For example, an application that demands very high accuracy needs a more distinctive biometric.
- What type of biometric is acceptable to the users? Different biometrics are acceptable in applications deployed in different demographics depending on the cultural, ethical,

social, religious, and hygienic standards of that society. The acceptability of a biometric in an application is often a compromise between the sensitivity of a community to various perceptions/taboos and the value/convenience offered by biometrics-based recognition.

A brief introduction to the most common biometrics is provided below [12, 16, 17].

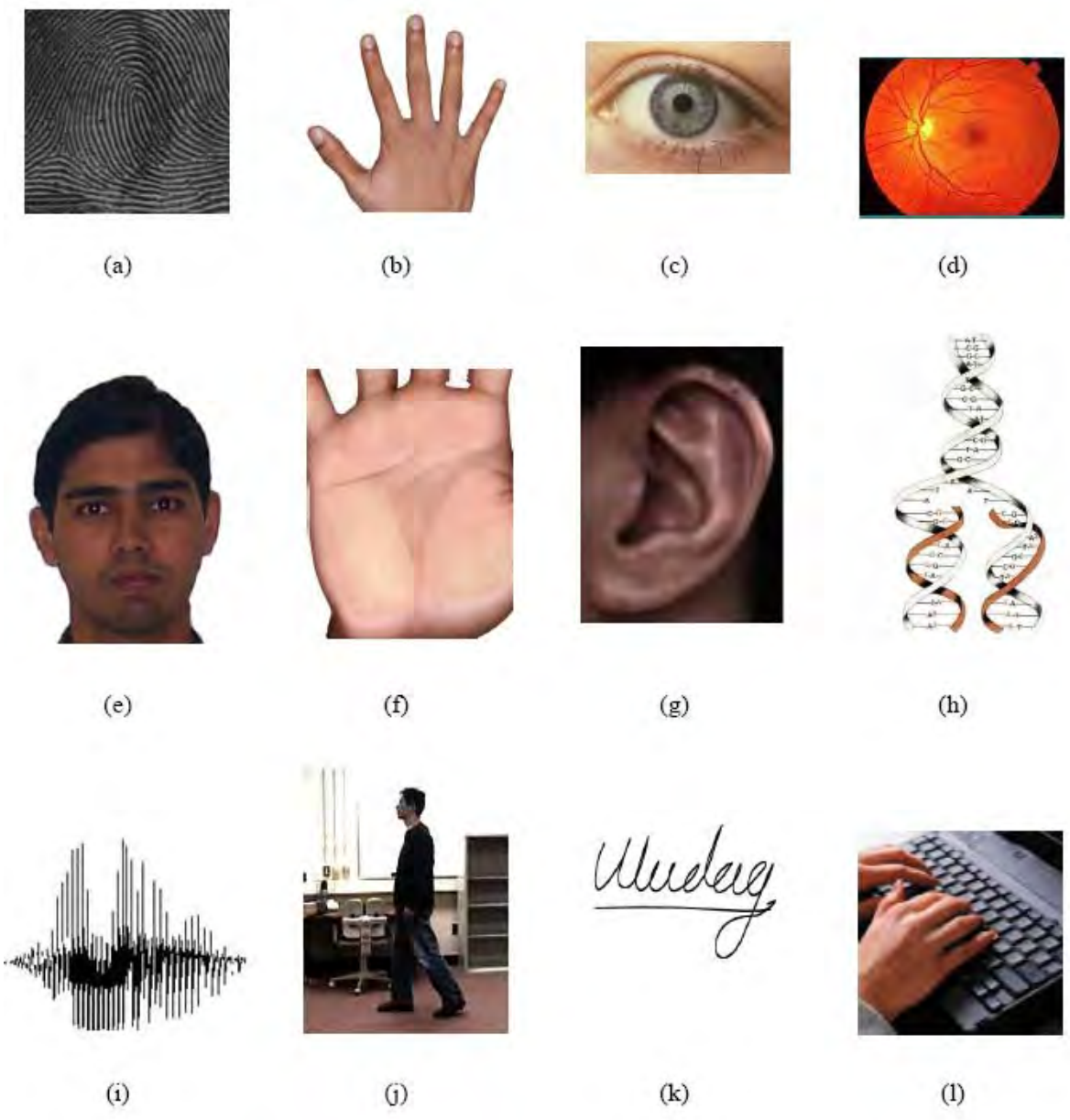


Figure 2-1: Characteristics that are being used for biometric recognition; (a) Fingerprint;(b) Hand-geometry; (c) Iris; (d) Retina; (e) Face; (f) Palmprint; (g) Ear structure; (h) DNA;(i) Voice; (j) Gait; (k) Signature and (l) Keystroke dynamics. [12].

***DNA*:**

DeoxyriboNucleic Acid (DNA) is the one-dimensional ultimate unique code for one's individuality, except for the fact that identical twins have identical DNA patterns. It is, however, currently being used mostly in the context of forensic applications for person recognition. Several issues limit the utility of this biometric for other applications:

i) Contamination and sensitivity: it is easy to steal a piece of DNA from an unsuspecting subject that can be subsequently abused for an ulterior purpose;

ii) Automatic real-time recognition issues: the present technology for DNA matching requires cumbersome chemical methods (wet processes) involving an expert's skills and is not geared for on-line non-invasive recognition;

iii) Privacy issues: information about susceptibilities of a person to a certain disease could be gained from the DNA pattern and there is a concern that the unintended abuse of genetic code information may result in discrimination, for example, in hiring practices.

***Ear*:**

It is known that the shape of the ear and the structure of the cartilaginous tissue of the pinna are distinctive. The features of an ear are not expected to be unique to an individual. The ear recognition approaches are based on matching the distance of salient points on the pinna from a landmark location on the ear.

***Face*:**

The face is one of the most acceptable biometrics because it is one of the most common methods of recognition that humans use in their visual interactions. In addition, the method of acquiring face images is non-intrusive. Facial disguise is of concern in unattended recognition applications. It is very challenging to develop facial recognition techniques that can tolerate the effects of aging, facial expressions, slight variations in the imaging environment, and variations in the pose of the face with respect to the camera (2D and 3D rotations).

***Gait*:**

Gait is the peculiar way one walks and is a complex spatio-temporal biometric. Gait is not supposed to be very distinctive, but is sufficiently characteristic to allow verification in some low-security applications. Gait is a behavioral biometric and may not stay invariant, especially over a large period of time, due to large fluctuations of body weight, major shift in the body weight, major injuries involving joints or brain. Acquisition of gait is similar to acquiring facial pictures and hence it may be an acceptable biometric. Because gait-based systems use video sequence footage of a walking person to measure several different movements of each articulate joint, it is computing and input intensive.

***Hand and finger geometry*:**

Some features related to a human hand (e.g., length of fingers) are relatively invariant and peculiar (although not very distinctive) to an individual. The image acquisition system requires cooperation of the subject and captures frontal and side view images of the palm flatly placed on a panel with outstretched fingers. The representational requirements of the hand are very small (typically less than 20 bytes), [44] which is an attractive feature for bandwidth- and memory-limited systems. Due to its limited distinctiveness, hand geometry-based systems are typically used for verification and do not scale well for identification applications. Finger geometry systems (which measure the geometry of only one or two fingers) may be preferred because of their compact size.

***Iris*:**

Visual texture of the human iris is determined by the chaotic morphogenetic processes during embryonic development and is posited to be distinctive for each person and each eye [19]. An iris image is typically captured using a non-contact imaging process. Capturing an iris image involves cooperation from the user, both to register the image of iris in the central imaging area and to ensure that the iris is at a predetermined distance from the focal plane of the camera. The iris recognition technology is believed to be extremely accurate and fast.

***Keystroke dynamics*:**

It is hypothesized that each person types on a keyboard in a characteristic way. This behavioral biometric is not expected to be unique to each individual but it offers sufficient discriminatory information to permit identity verification. Keystroke dynamics is a behavioral biometric; for some individuals, one may expect to observe large variations from typical typing patterns. The keystrokes of a person could be monitored by using a system unobtrusively as that person is keying in information.

***Retinal scan*:**

The retinal vasculature is rich in structure and is supposed to be a characteristic of each individual and each eye. It is claimed to be the most secure biometric since it is not easy to change or replicate the retinal vasculature. The image capture requires a person to peep into an eyepiece and focus on a specific spot in the visual field so that a predetermined part of the retinal vasculature may be imaged. The image acquisition involves cooperation of the subject, entails contact with the eyepiece, and requires a conscious effort on the part of the user. All these factors adversely affect public acceptability of retinal biometrics. Retinal vasculature can reveal some medical conditions (e.g., hypertension), which is another factor standing in the way of public acceptance of retinal scan-based biometrics.

***Signature*:**

The way a person signs his name is known to be a characteristic of that individual. Although signatures require contact and effort with the writing instrument, they seem to be acceptable in many government, legal, and commercial transactions as a method of verification. Signatures are a behavioral biometric that change over a period of time and are influenced by physical and emotional conditions of the signatories. Signatures of some people vary a lot: even successive impressions of their signature are significantly different. Furthermore, professional forgers can reproduce signatures to fool the unskilled eye.

***Voice*:**

Voice capture is unobtrusive and voice print is an acceptable biometric in almost all societies. Voice may be the only feasible biometric in applications requiring person recognition over a telephone. Voice is not expected to be sufficiently distinctive to permit identification of an individual from a large database of identities. Moreover, a voice signal available for recognition is typically degraded in quality by the microphone, communication channel, and digitizer characteristics. Voice is also affected by a person's health (e.g., cold), stress, emotions, and so on. Besides, some people seem to be extraordinarily skilled in mimicking others.

2.4 Biometric Systems

Identity management refers to the challenge of providing authorized users with secure and easy access to information and services across a variety of networked systems. A reliable identity management system is a critical component in several applications that render their services only to legitimate users. Examples of such applications include physical access control to a secure facility, e-commerce, access to computer networks and welfare distribution. The primary task in an identity management system is the determination of an individual's identity. Traditional methods of establishing a person's identity include knowledge-based (e.g., passwords) and token-based (e.g., ID cards) mechanisms. These surrogate representations of the identity can easily be lost, shared or stolen. Therefore, they are not sufficient for identity verification in the modern day world. Biometrics offers a natural and reliable solution to the problem of identity determination by recognizing individuals based on their physiological and/or behavioral characteristics that are inherent to the person.

A typical biometric system consists of four main modules. The sensor module is responsible for acquiring the biometric data from an individual. The feature extraction module processes the acquired biometric data and extracts only the salient information to form a new representation of the data. Ideally, this new representation should be unique for each person and also relatively invariant with respect to changes in different samples of the same biometric collected from the same person. The matching module compares the extracted feature set with the templates stored in the system database and determines

the degree of similarity (dissimilarity) between the two. The decision module either verifies the identity claimed by the user or determines the user's identity based on the degree of similarity between the extracted features and the stored template(s).

Biometric systems can provide three main functionalities, namely, (i) verification, (ii) identification and (iii) negative identification. Figure 2-2 shows the flow of information in verification and identification systems. In verification or authentication, the user claims an identity and the system verifies whether the claim is genuine. For example, in an ATM application, the user may claim a specific identity, say John Doe, by entering his Personal Identification Number (PIN). The system acquires the biometric data from the user and compares it only with the template of John Doe. Thus, the matching is 1:1 in a verification system. If the user's input and the template of the claimed identity have a high degree of similarity, then the claim is accepted as "genuine". Otherwise, the claim is rejected and the user is considered an "impostor". In short, a biometric system operating in the verification mode, answers the question "Are you who you say you are?"

In a biometric system used for identification, the user does not explicitly claim an identity. However, the implicit claim made by the user is that he is one among the persons already enrolled in the system. In identification, the user's input is compared with the templates of all the persons enrolled in the database and the identity of the person whose template has the highest degree of similarity with the user's input is output by the biometric system. Typically, if the highest similarity between the input and all the templates is less than a fixed minimum threshold, the system outputs a reject decision which implies that the user presenting the input is not one among the enrolled users. Therefore, the matching is 1:*N* in an identification system. An example of an identification system could be access control to a secure building. All users who are authorized to enter the building would be enrolled in the system. Whenever a user tries to enter the building, he presents his biometric data to the system and upon determination of the user's identity, the system grants him the preset access privileges. An identification system answers the question "Are you really someone who is known to the system?"

Negative identification systems are similar to identification systems because the user does not explicitly claim an identity. The main factor that distinguishes the negative identification functionality from identification is the user's implicit claim that he is *not* a person who is already enrolled in the system. Negative identification is also known as screening. As in identification, the matching is 1:*N* in screening. However in screening, the system will output an identity of an enrolled person only if that person's template has the highest degree of similarity with the input among all the templates and if the corresponding similarity value is greater than a fixed threshold. Otherwise, the user's claim that he is not already known to the system is accepted. Screening is often used at airports to verify whether a passenger's identity matches with any person on a "watch-list". Screening can also be used to prevent the issue of multiple credential records (e.g., driver's license, passport) to the same person. To summarize, negative identification answers the question "Are you who you say you are not?"

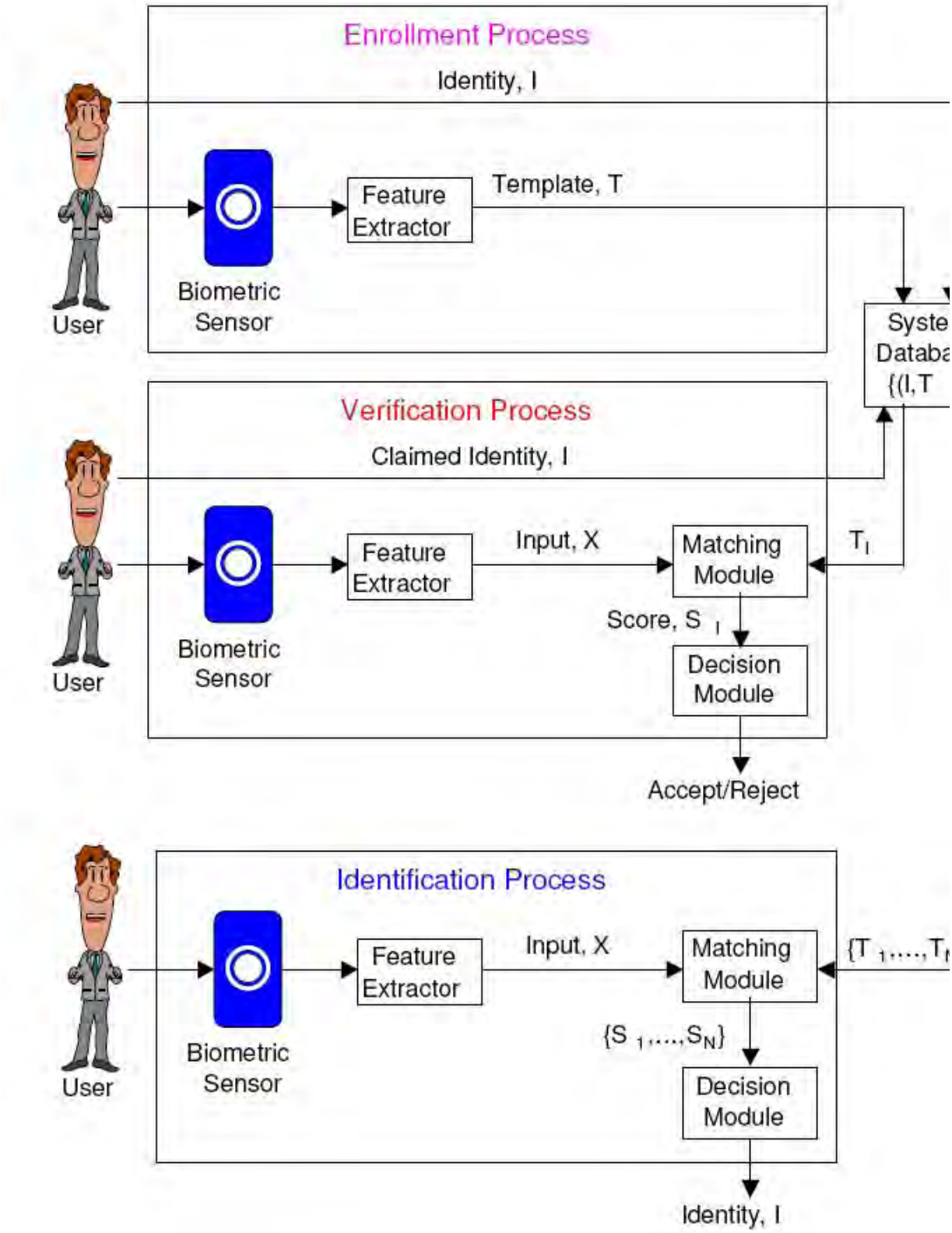


Figure 2-2: Information flow in biometric systems.

Verification functionality can be provided by traditional methods like passwords and ID cards as well as by biometrics. The negative identification functionality can be provided only by biometrics. Further, biometric characteristics are inherent to the person whose identity needs to be established. Hence, they cannot be lost, stolen, shared, or forgotten. Therefore, biometric traits provide more security than traditional knowledge-based or token-based identification methods. They also discourage fraud and eliminate the possibility of repudiation. Finally, they are more convenient to use because they eliminate the need for remembering multiple complex passwords and carrying identification cards. Although biometric systems have some limitations, they offer a number of advantages over traditional security methods and this has led to their widespread deployment in a variety of civilian applications. Figure 2-3 shows some examples of biometrics deployment in civilian applications.

Figure 2-3: *Some illustrations of deployment of biometrics in civilian applications; (a) A fingerprint verification system manufactured by Digital Personal Inc. used for computer and network login; (b) An iris-based access control system at the Umea airport in Sweden that verifies the frequent travelers and allow them to flights; (c) A cell phone manufactured by LG Electronics that recognizes authorized users using fingerprints (sensors manufactured by Authentec Inc.) and allow them to the phone's special functionalities such as mobile-banking; (d) The US-VISIT immigration system based on fingerprint and FR technologies and (e) A hand geometry system at Disney World that verifies seasonal and yearly pass-holders to allow them fast entry.*

CHAPTER 3 ARTIFICIAL NEURAL NETWORKS

3.1 Introduction

ANN have been successfully applied to problems in pattern classification, function approximation, optimization, pattern matching and associative memories [5, 7, 8]. ANN models or simply “neural networks” known by many names such as neuro-computers, Parallel Distributed Processing (PDP) models, neuromorphic systems, layered self-adaptive networks, and connectionist models [7, 9]. Here, we use the name ANN and we use these networks as a vehicle for adaptively developing the coefficients of decision functions via successive presentations of training sets of patterns. They have been studied for many years in the hope of achieving human-like performance in the fields of speech and image recognition [7]. They are used in pattern recognition systems as pattern classifiers [7, 10]. These models are composed of many non-linear computational elements operating in parallel and arranged in patterns reminiscent to Biological Neural Networks (BNN) based on our present understanding of the biological nervous systems [7]. The computational elements or nodes are connected via weights that are typically adapted during training. Thus, instead of performing a set of program instructions sequentially as in a Von Neumann computer architecture, neural network models explore many competing hypotheses simultaneously by using massively parallel networks composed of many computational elements connected by links with variable or adaptable weights [7].

They have the greatest potential in areas such as speech and image recognition where many hypotheses are pursued in parallel, high computations are required, and the current best systems are far from equaling human performance [6, 7, 10].

The potential benefits of neural networks extend beyond the high computation rates provided by massive parallelism. Neural networks typically provide a greater degree of robustness or fault tolerance than sequential algorithms, because they have many processing nodes each with primarily local connections. Few erroneous training data or damage to a few nodes or links thus need not impair overall performance significantly [6, 7, 10].

Adaptation or learning where the training data is limited, such as in speech and image recognition, is a major focus of neural network research [6, 7, 10]. Adaptation also provides a degree of robustness by compensating for minor variabilities in characteristics of processing elements.

Neural network models are specified by the network technologies or architectures, node characteristics, and training or learning rules. These rules specify an initial set of weights and indicate how weights should be adapted during training. Both design procedures and training rules are the topics of much current research. The ANN is trained to classify a given task in a supervised or unsupervised learning mode, depending on the particular architecture [7, 9].

3.2 Biological Motivation

Studying the real BNN leads to new insights and algorithmic improvements in designing and implementing ANN. Studies over the past few decades have shed some light on the construction and operation of our brain and nervous systems [7].

The basic building block of the nervous system is the nervous cell or neuron whose main function is to conduct excitation. The major components of a neuron as shown in the Figure 3-1 includes a cell body, dendrites or ramifying fibers conveying impulses toward this body, and the axon which conducts impulses away from the cell body to other neurons [7, 9].

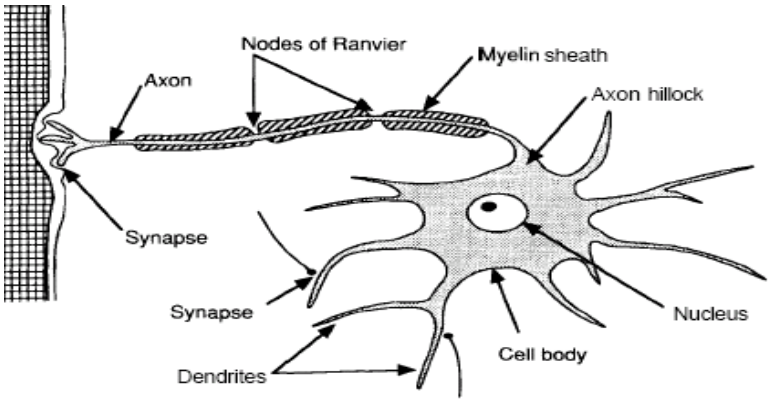


Figure 3-1: The major structure of a typical nerve cell [7].

Figure 3-1, a conceptual diagram of a neuron, is a sketch of only one representation of a neuron. There are many kinds of neurons, with different configurations and functions. The neuron in this figure probably resembles a motor neuron more than most other types, such as sensory neurons, but it is meant only to convey the basic configuration and technology. Note that the signal flow goes from right to left, from dendrites, through the cell body, and out through the axon. The signal from one neuron is passed onto another by means of a connection between the axon of the first and dendrite of the second. This connection is called a synapse. Axons are connected onto the trunk of a dendrite, but they can also be connected directly onto the cell body. The synapse is believed to play the key role in the mechanism responsible for the establishment of new links in the nervous system. It is presumed that the establishment of such links is accompanied by certain changes (chemical or structural) in the synapses which transmit impulses in a definite direction. According to the physiological theories of memory, and nervous impulse passing through a definite group of neurons, leaves behind it a physical trace in the form of electrical and mechanical changes of synapses. Such changes facilitate the secondary passage of the impulse along the trial that has been blazed [7, 9].

The waves evolving in the brain are electromagnetic oscillations of different frequencies [7]. The lowest frequencies correspond to a state of response when a person is relaxed and sitting with his eyes closed. Once he receives an assignment, for instance, to do a sum, the curve of his biological currents immediately changes and exhibits signs of much higher frequencies [7].

The human brain is composed of many different parallel, distributed systems, performing well defined functions, but under the control of a serial-processing system at one or more levels. It has a large number of neurons with typical estimates of in the order of 10-500 billions. According to one estimate by Stubbs, neurons are arranged into about 1000 main modules, each with about 500 bio-neural networks. Each network has on the order of 100,000 neurons. The axon of each neuron connects to about 100 (but sometimes several thousands) other neurons, and this value varies greatly from neuron to neuron and from neuron type to neuron type [7, 9].

3.3 ANN Applications

There are several types of neural network architectures that are in use today. They include the BPN (Back Propagation Network), CPN (Counter Propagation Network), the Madeline, the neocognition and so on. Some, such as the BPN, are for general purpose. Some potential areas in which they are implemented include character recognition, speech and image recognition, medical diagnosis, human face and finger print recognition. In some of these areas, they may be implemented as hybrid with other systems such as expert systems and artificial intelligence [5, 6, 7].

An expert system is a software based system that describes the behavior of an expert in some field by capturing the knowledge of one or more experts in the form of rules and symbols. It is more efficient than neural networks in problems with inadequate data, which might not be enough to train a neural network [7, 9].

3.3.1 Advantages of Neural Computing

There are a variety of benefits that an analyst realizes from using neural networks in their work.

- Pattern recognition is a powerful technique for harnessing the information in the data and generalizing about it. Neural networks learn to recognize the patterns which exist in the data set.
- The system is developed through learning rather than programming. Programming is much more time consuming for the analyst and requires the analyst to specify the exact behavior of the model. Neural networks teach themselves the patterns in the data freeing the analyst for more interesting work.
- Neural networks are flexible in a changing environment. Rule based systems or programmed systems are limited to the situation for which they were designed—when conditions change, they are no longer valid. Although neural networks may take some time to learn a sudden drastic change, they are excellent at adapting to constantly changing information.

- Neural networks can build informative models where more conventional approaches fail. Because neural networks can handle very complex interactions they can easily model data which is too difficult to model with traditional approaches such as inferential statistics or programming logic.
- Performance of neural networks is at least as good as classical statistical modeling and better on most problems. The neural networks build models that are more reflective of the structure of the data in significantly less time.
- Neural networks now operate well with modest computer hardware. Although neural networks are computationally intensive, the routines have been optimized to the point that they can now run in reasonable time on personal computers. They do not require supercomputers as they did in the early days of neural network research.

3.3.2 Limitations of Neural Computing

There are some limitations to neural computing. The key limitation is the neural network's inability to explain the model it has built in a useful way. Analysts often want to know why the model is behaving as it is. Neural networks get better answers but they have a hard time explaining how they got there [11].

There are a few other limitations that should be understood. First, it is difficult to extract rules from neural networks. This is sometimes important to people who have to explain their answer to others and to people who have been involved with artificial intelligence, particularly expert systems which are rule-based.

As with most analytical methods, you cannot just throw data at a neural network and get a good answer. You have to spend time understanding the problem or the outcome you are trying to predict. And, you must be sure that the data used to train the system are appropriate and are measured in a way that reflects the behavior of the factors. If the data are not representative of the problem, neural computing may not produce good results. This is a classic situation where "garbage in" will certainly produce "garbage out."

Finally, it can take time to train a model for a very complex data set. Neural techniques are computational intensive and will be slow on low end PCs or machines without math coprocessors. It is important to remember though that the overall time to get results can still be faster than other data analysis approaches, even when the system takes longer to train.

3.4 Network Architecture

3.4.1 The Neuron or Processing Element (PE)

The individual computational element that make up the artificial neural system models are referred to as node, units, processing elements (PE) or artificial neurons (rarely called since they are only crude models of their biological counterpart) [7, 9]. Figure 3-2 shows the complete Adaline consisting of the adaptive linear combiner, in the dashed box, and a bipolar output function.

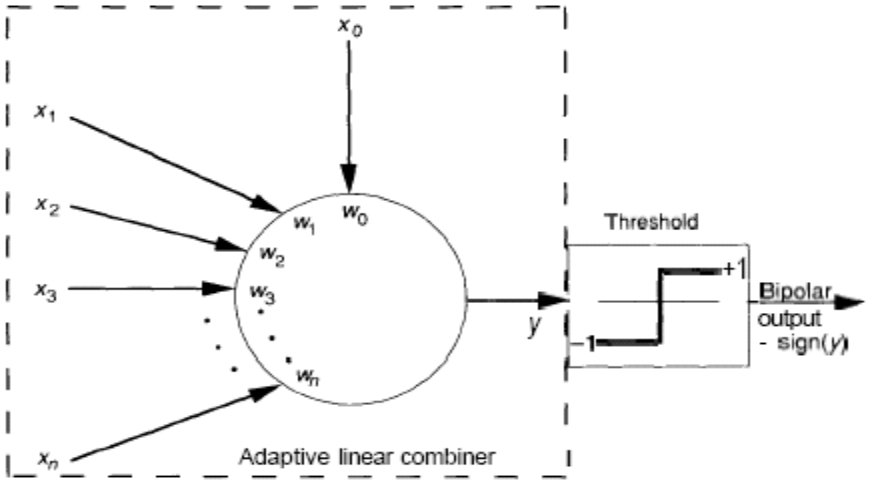


Figure 3-2: The complete Adaline [7].

The artificial neuron or PE is the basic or primitive building block of a neural network. Like a real neuron, the PE has many inputs but has only a single output which can fan out to many other PEs in an ANN. Each input to the PE has associated with it a quantity called weight or connection strength. All these quantities have analogues in the standard biological neuron model, i.e., the output of the PE corresponds to the firing frequency of

the biological neuron, and the weight corresponds to the strength of the synaptic weight between neurons. In our models, these quantities will be represented as real numbers. An input connection may be excitory, having positive weights, or inhibitory, having negative weights and both types are usually considered together constituting the most common forms of input to a PE.

For the BPN, the PE used is called the Adaline (Adaptive Linear Neuron) which consists of the ALC (Adaptive Linear Combiner) and an output activation function. The ALC has many weighted inputs, an optional bias term, and a combiner or summer unit. Its input may be continuous (normalized or not) or binary and the weights or connection strength via which the inputs are linked to the summer may be positive or negative real numbers. The bias term is a weight on a connection that has its input value always equal to one and its inclusion is largely a matter of experience in helping learning convergence during training [7, 9].

The output function thresholder may be a hard limiter, a linear output, a sigmoidal function, a hyperbolic tangent function or another function depending on the problem to be solved. Figure 3-2 shows the complete PE or artificial neuron structure with different output functions.

The PE performs a sum of products calculation using the input and weight vectors and applies an output function to get a single output value. Using the notation in Figure 3-2,

$$y = w_0 + \sum_{j=1}^{n} w_j x_j \qquad 3.1$$

Where w_0 is the bias weight and n is the number of input nodes. If we make the identification, $x_0=1$, we can rewrite the preceding equation as

$$y = \sum_{j=0}^{n} w_j x_j \qquad 3.2$$

Or in vector notation

$$y = w^T x \qquad 3.3$$

Once the net input is calculated, it is converted to an activation value or simply activation for the PE. Since in ALC (as well as in the majority of neuron models) the activation and the net input are the same, the two terms are used interchangeably. We can thus determine the output value i of the PE by applying an output function [7]:

$$i = f(y) \qquad 3.4$$

The PE, Adaline or the ALC is adaptive in the sense that there exists a well-defined procedure for modifying the weights in order to allow the device to give correct output value for the given input and this weight adjusting scheme is dealt in Appendix B.

3.4.2 The Back Propagation Network (BPN)

An ANN that is found to be useful in addressing problems requiring recognition of complex patterns and performing nontrivial mapping functions is the BPN, formalized first by Werbos, and later by Parker and Rummelhart and McClelland. This network is designed to operate as a multilayer, feedforward network, using the supervised mode of learning.

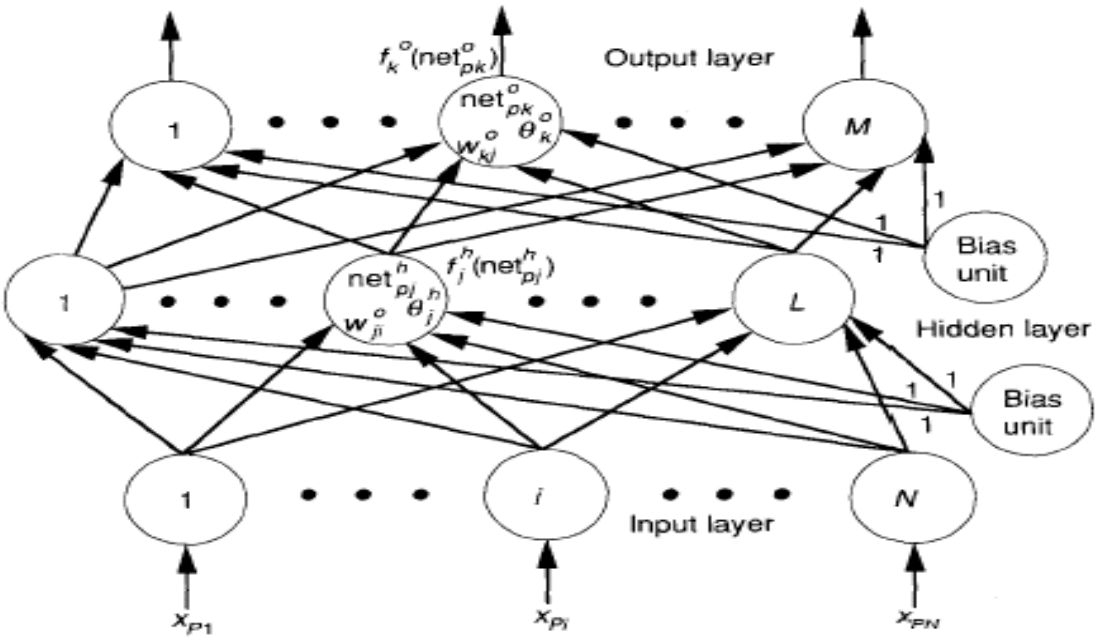


Figure 3-3: Three-layer BPN architecture [7].

The typical BPN has an input layer, an output layer, and at least one hidden layer. There is no theoretical limit on the number of hidden layers but typically there is just one or two. Each layer is fully connected to the succeeding layer, as shown in Figure 3-3. This architecture has spawned a large class of network types with many different topologies and training methods.

The formal mathematical description of BPN operation is presented with a detailed derivation of the generalized delta rule which is the learning algorithm of the network in Appendix B.

3.5 Practical Considerations

3.5.1 Training Data

It is essential to have some definition regarding the selection of number of training-vector pairs for the BPN. Unfortunately, there is no single definition that applies to all cases. As with many aspects of neural-network systems, experience is often the best teacher. As you gain facility with using networks, you will also gain an appreciation for how to select and prepare training sets [7, 9].

In general, you can use as many data as you have to train the network, although you may not need to use them all. From the available training data, a small subset is often all that you need to train a network successfully. The remaining data can be used to test the network to verify that the network can perform the desired mapping on input vectors it has never encountered during training [7, 9].

If you are training a network to perform in a noisy environment, such as the pixel-image-to-ASCII example, then include some noisy input vectors in the data set. Sometimes the addition of noise to the input vectors during training helps the network to converge even if no noise is expected on the inputs [7, 9].

The BPN is good at generalization. What we mean by generalization here is that, given several different input vectors, all belonging to the same class, a BPN will learn to key

off of significant similarities in the input vectors. Irrelevant data will be ignored. As an example, suppose we want to train a network to determine whether a bipolar number of length 5 is even or odd. With only a small set of examples used for training, the BPN will adjust its weights so that a classification will be made solely on the basis of the value of the least significant bit in the number: The network learns to ignore the irrelevant data in the other bits [7, 9].

In contrast to generalization, the BPN will not extrapolate well. If a BPN is inadequately or insufficiently trained on a particular class of input vectors, subsequent identification of members of that class may be unreliable. Make sure that the training data cover the entire expected input space. During the training process, select training-vector pairs randomly from the set, if the problem lends itself to this strategy. In any event, do not train the network completely with input vectors of one class, and then switch to another class: The network will forget the original training [7, 9].

If the output function is sigmoidal, then you will have to scale the output values. Because of the form of the sigmoidal function, the network outputs can never reach 0 or 1. Therefore, use values such as 0.1 and 0.9 to represent the smallest and largest output values. You can also shift the sigmoid so that, for example, the limiting values become ±0.4. Moreover, you can change the slope of the linear portion of the sigmoid curve by including a multiplicative constant in the exponential. There are many such possibilities that depend largely on the problem being solved [7, 9].

3.5.2 Network Sizing

Just how many nodes are needed to solve a particular problem? Are three layers always sufficient? As with the questions concerning proper training data, there are no strict answers to questions such as these. Generally, three layers are sufficient. Sometimes, however, a problem seems to be easier to solve with more than one hidden layer. In this case, easier means that the network learns faster [7].

The size of the input layer is usually dictated by the nature of the application. You can often determine the number of output nodes by deciding whether you want analog values or binary values on the output units [7].

Determining the number of units to use in the hidden layer is not usually as straightforward as it is for the input and output layers. The main idea is to use as few hidden-layer units as possible, because each unit adds to the load on the CPU during simulation. Of course, in a system that is fully implemented in hardware (one processor per processing element), additional CPU loading is not as much of a consideration (interprocessor communication may be a problem, however).

Paper [7] hesitates to offer specific guidelines except to say that, in their experience, for networks of reasonable size (hundreds or thousands of inputs), the size of the hidden layer needs to be only a relatively small fraction of that of the input layer. If the network fails to converge to a solution it may be that more hidden nodes are required. If it does converge, one might try fewer hidden nodes and settle on a size on the basis of overall system performance.

It is also possible to remove hidden units that are superfluous. If you examine the weight values on the hidden nodes periodically as the network trains, you will see that weights on certain nodes change very little from their starting values. These nodes may not be participating in the learning process, and fewer hidden units may suffice. There is also an automatic method, developed by Rumelhart, for pruning unneeded nodes from the network [Unscheduled talk given at the Second International Conference on Neural Networks, San Diego, June 1988] [7].

3.5.3 Weights and Learning Parameters

Weights should be initialized to small, random values - say between ±0.5 as should the bias terms which are treated just like a weight, and it participates in the learning process as a weight. Another possibility is simply to remove the bias term altogether; its use is optional. Selection of a value for the learning rate parameter, η, has a significant effect on

the network performance. Usually, η must be a small number on the order of 0.05 to 0.5 to ensure that the network will settle to a solution [7, 9].

A small value of η; means that the network will have to make a large number of iterations, but that is the price to be paid. It is often possible to increase the size of η as learning proceeds. Increasing η as the network error decreases will often help to speed convergence by increasing the step size as the error reaches a minimum, but the network may bounce around too far from the actual minimum value if η gets too large [7, 9].

Another way to increase the speed of convergence is to use a technique called momentum. When calculating the weight-change value, $\Delta_p w$, we add a fraction of the previous change. This additional term tends to keep the weight changes going in the same direction, hence the term momentum. The weight change equations on the output layer then become like [7, 9]

$$w^o_{kj}(t + 1) = w^o_{kj}(t) + \eta\delta^o_{pk}\, i_{pj} + \alpha\Delta_p w^o_{kj}(t - 1) \qquad 3.5$$

With a similar equation on the hidden layer, In Eq. (3.5), α is the momentum parameter, and it is usually set to a positive value less than 1. The use of the momentum term is optional [7, 9].

A final topic concerns the possibility of converging to a local minimum is weight space. Figure 3-4 illustrates the idea. Once a network settles on a minimum, whether local or global, learning ceases. If a local minimum is reached, the error at the network outputs may still be unacceptably high. Fortunately, this problem does not appear to cause much difficulty in practice. If a network stops learning before reaching an acceptable solution, a change in the number of hidden nodes or in the learning parameters will often fix the problem; or we can simply start over with a different set of initial weights. When a network reaches an acceptable solution, there is no guarantee that it has reached the global minimum rather than a local one. If the solution is acceptable from an error standpoint, it does not matter whether the minimum is global or local or even whether the training was halted at some point before a true minimum was reached [7, 9].

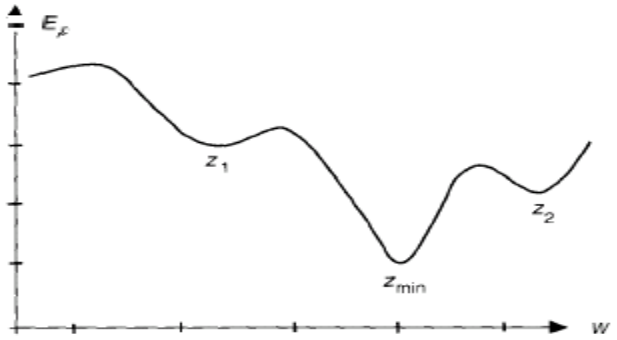


Figure 3-4: A cross-section of a hypothetical error surface in weight space [7].

On the Figure 3-4, the point, z_{min}, is called the global minimum. Notice, however, that there are other minimum points, z_1 and z_2. A gradient-descent search for the global minimum might accidentally find one of these local minima instead of the global minimum [7].

CHAPTER 4 FACE RECOGNITION

4.1 Introduction

Within the field of computer vision, a considerable amount of research has been performed in recent times on automated methods for recognizing the identity of individuals from their facial images. The major motivating factors for this are the understanding of human perception, and a number of security and surveillance applications such as access to ATMs, airport security, tracking of individuals and law enforcement.

Computational models of FR must address several difficult problems. This difficulty arises from the fact that faces must be represented in a way that best utilizes the available face information to distinguish a particular face from all other faces. Faces pose a particularly difficult problem because all faces are similar to one another in that they contain the same set of features such as eyes, nose, mouth arranged in roughly the same manner.

In Figure 4-1, the outline of a typical FRS is given. This outline heavily carries the characteristics of a typical pattern recognition system.

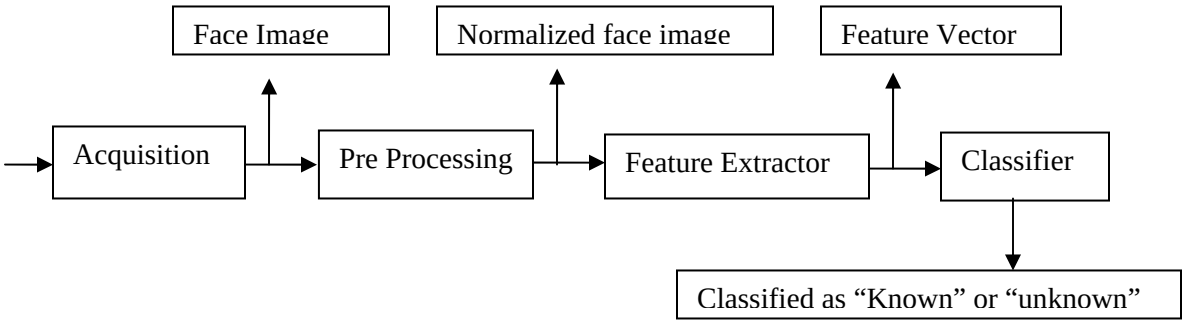


Figure 4-1: Outline of a typical FRS. [35]

Acquisition module: This is the entry point of the FR process. It is the module where the face image under consideration is presented to the system. An acquisition module can request a face image from several different environments.

Pre-processing module: By means of early vision techniques, face images are normalized and if desired, they are enhanced to improve the recognition performance of the system. Some or all of the following pre-processing steps may be implemented in a FRS:

- Image size normalization
- Illumination normalization
- High-pass filtering
- Background removal
- Translational and rotational normalizations

Feature extraction module: After performing some pre-processing (if necessary) techniques, the normalized face images are presented to the feature extraction module in order to find the key features that are going to be used for classification.

Classification module: In this module, with the help of a pattern classifier, extracted features of the face image is compared with the ones stored in a face library (or face database). After doing this comparison, face image is classified as either known or unknown.

Most biometric techniques are inconvenient due to the necessity of interaction with the individual who is to be identified or authenticated. FR on the other hand can be a non-intrusive technique. This is one of the reasons why this technique has caught an increased interest from the scientific community. FR holds several advantages over other biometric techniques. It is natural, non-intrusive and easy to use. In a study considering the compatibility of six biometric techniques (face, finger, hand, voice, eye, and signature) with machine readable travel documents (MRTD) [20], FR scored the highest percentage of compatibility. The corresponding graph is shown in Figure 4-2.

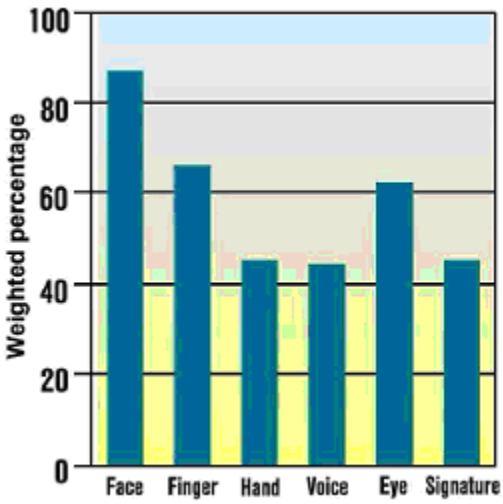


Figure 4-2: Comparison of machine readable travel documents (MRTD) compatibility with six biometric techniques; face, finger, hand, voice, eye, signature. [20].

The increased interest in automated FRS have gained, from environments other than the scientific community is largely due to increasing public concerns for security, especially due to the many events of terror around the world after September 11th 2001 [35].

4.2 Face Recognition Tasks

Face Verification (”Am I who I say I am?”) is a one-to-one match that compares a query face image against a template face image whose identity is being claimed. To evaluate the verification performance, the verification rate (the rate at which legitimate users is granted access) vs. false accept rate (the rate at which imposters are granted access) is plotted, called ROC curve. A good verification system should balance these two rates based on operational needs.

Face Identification (”Who am I?”) is a one-to-many matching process that compares a query face image against all the template images in a face database to determine the identity of the query face. The identification of the test image is done by locating the image in the database that has the highest similarity with the test image. The identification process is a”closed” test, which means that the sensor takes an observation of an individual that is known to be in the database. The test subject’s (normalized) features are compared to the other features in the system’s database and a similarity score

is found for each comparison. These similarity scores are then numerically ranked in a descending order. The percentage of times that the highest similarity score is the correct match for all individuals is referred to as the"top match score." If any of the top *r* similarity scores corresponds to the test subject, it is considered as a correct match in terms of the cumulative match. The percentage of times one of those *r* similarity scores is the correct match for all individuals is referred to as the"Cumulative Match Score". The" Cumulative Match Score" curve is the rank *n* versus percentage of correct identification, where rank *n* is the number of top similarity scores reported.

The watch list ("Are you looking for me?") method is an open-universe test. The test individual may or may not be in the system database. That person is compared to the others in the system's database and a similarity score is reported for each comparison. These similarity scores are then numerically ranked so that the highest similarity score is first. If a similarity score is higher than a preset threshold, an alarm is raised. If an alarm is raised, the system thinks that the individual is located in the system's database. There are two main items of interest for watch list applications. The first is the percentage of times the system raises the alarm and it correctly identifies a person on the watch list. This is called the"Detection and Identification Rate." The second item of interest is the percentage of times the system raises the alarm for an individual that is not on the watch list (database). This is called the" False Alarm Rate."

4.3 Face Recognition by Humans

When building artificial FRS, scientists try to understand the architecture of human FRS. Focusing on the methodology of human FRS may be useful to understand the basic system. However, the human FRS utilizes more than that of the artificial FRS which is just 2-D data. The human FRS uses some data obtained from some or all of the senses; visual, auditory, tactile, etc. All these data are used either individually or collectively for storage and remembering of faces. In many cases, the surroundings also play an important role in human FRS. It is hard for a machine recognition system to handle so much of data and their combinations. However, it is also hard for a human to remember many faces due to storage limitations. A key potential advantage of a machine system is

its memory capacity [21], whereas for a human FRS the important feature is its parallel processing capacity.

The issue of “which features humans use for FR” has been studied and it has been argued that both global and local features are used for FR. It is harder for humans to recognize faces which they consider as neither “attractive” nor “unattractive”.

If there are dominant features present such as big ears, a small nose, etc. holistic descriptions may not be used. Also, recent studies show that an inverted face (i.e. all the intensity values are subtracted from 255 to obtain the inverse image in the grey scale) is much harder to recognize than a normal face. Hair, eyes, mouth, face outlines have been determined to be more important than nose for perceiving and remembering faces. It has also been found that the upper part of the face is more useful than the lower part of the face for recognition. Also, aesthetic attributes (e.g. beauty, attractiveness, pleasantness, etc.) play an important role in FR; the more attractive faces are easily remembered.

For humans, photographic negatives of faces are difficult to recognize. But, not much study was done on why it is difficult to recognize negative images of human faces. Also, a study on the direction of illumination [24] showed the importance of top lighting; it is easier for humans to recognize faces illuminated from top to bottom than the faces illuminated from bottom to top.

According to the neurophysicists, the analysis of facial expressions is done in parallel to FR in human FRS. Some prosopagnosic patients, who have difficulties in identifying familiar faces, seem to recognize facial expressions due to emotions. Patients who suffer from organic brain syndrome do poorly at expression analysis but perform FR quite well.

4.4 Machine Recognition of Faces

Although studies on human FR were expected to be a reference for machine recognition of faces, research on machine recognition of faces has developed independent of studies on human FR. During 1970’s, typical pattern classification techniques used measurements between features in faces or face profiles [23]. During 1980’s, work on FR

remained nearly stable. Since the early 1990's, research interest on machine recognition of faces has grown tremendously.

The basic question relevant for face classification is that; what form the structural code (for encoding the face) should take to achieve FR. Two major approaches are used for machine identification of human faces; geometrical local feature based methods, and holistic template matching based systems. Also, combinations of these two methods, namely hybrid methods are used. The first approach, the geometrical local feature based one, extracts and measures discrete local features (such as eye, nose, mouth, hair, etc.) for retrieving and identifying faces. Then, standard statistical pattern recognition techniques and/or neural network approaches are employed for matching faces using these measurements [23]. One of the well known geometrical-local feature based methods is the Elastic Bunch Graph Matching (EBGM) technique.

The other approach, the holistic one, conceptually related to template matching, attempts to identify faces using global representations [22]. Holistic methods approach the face image as a whole and try to extract features from the whole face region. In this approach, as in the previous approach, the pattern classifiers are applied to classify the image after extracting the features. One of the methods to extract features in a holistic system is applying statistical methods such as PCA to the whole image. PCA can also be applied to a face image locally; in that case the approach is not holistic.

Whichever method is used, the most important problem in FR is the curse of dimensionality problem. Appropriate methods should be applied to reduce the dimension of the studied space. Working on higher dimension causes overfitting where the system starts to memorize. Also, computational complexity would be an important problem when working on large databases.

4.5 Face Acquisition

This section looks at how the original data is acquired before the issue of representation is raised. How many and what type of face images are needed? How much and what kind of variation should be present in the images?

4.5.1 Face Databases

The environment and manner in which a database of face images is collected is vital to the success of any FRS in which it is used. Almost without exception, FR research has been carried out with highly constrained data, with variations due to lighting, expression and pose either fixed or within unrealistic limits (if compared to variations encountered in real life data).

Two UK databases are available for FR task. First, the ORL face database is a small database of 40 people (400 images) showing some pose, with lighting and expression variations. The usefulness of the ORL database lies in having a large number of comparative results from different groups. Second, the Manchester face database (MFD) is larger (30 people, with 690 images). The training and test data of MFD have been deliberately kept separate to prevent systems from using spurious environmental details, such as lighting or background features, to classify individuals, and have at least 3 weeks between their collections for each person to introduce more realistic variability into the data. The training images have fairly constrained lighting, whereas the test images are more variable. Two levels of difficulty are present in test data, with different levels of variability. The first is fairly easy as the images are quite similar to the training images, only allowing changes to hairstyle, background and the wearing of glasses. The second is much harder, featuring occlusion with hands, dark glasses and covered hair. Although the MFD tests FR methods more thoroughly than the ORL database, it does not have so many published comparative results.

The largest collection of face images to date is the FERET (FacE REcognition Technology) database currently still under development by the US Army Research Laboratory. This has made great advances in constructing a standard by which competing FRS can be compared by conducting independent testing of leading algorithms. They have allowed pose movements from frontal to profile and limited lighting variations within the group of images for each person, and have collected the data over a period of time to allow changes in the person's appearance, clothing and lighting. The database

evolves from year to year. Its major disadvantage is its unavailability to non-US institutions.

4.6 Face Representation

For a FRS to perform effectively, it is important to isolate and extract the salient features in the input data to represent the face in the most efficient way. The abstract element of such a representation can be made up in a variety of ways and which approach will be appropriate depends on the task.

One of the main problems in computer vision, especially in FR, is dimensionality reduction to remove much of the redundant information in the original images. Simple mechanisms, such as sub-sampling, may give a rough reduction, but use of more specific and sophisticated preprocessing techniques to an image is still required for the best results.

4.6.1 Principal Component Analysis and 'Eigenfaces'

PCA is a simple statistical dimensionality reducing technique that has perhaps become the most popular for FR. PCA, via the Kahunen-Loeve transform, can extract the most statistically significant information for a set of images as a set of eigenvectors (usually called ‘eigenfaces’ when applied to faces), which can be used both to recognize and reconstruct face images. Eigenvectors can be regarded as a set of generalized features which characterize the image variations in the database. Once the face images are normalized, they can be treated as 1-D array of pixel values. Each image has an exact representation via a linear combination of these eigenvectors and an arbitrarily close approximation using the most significant eigenvectors (that is, those with the highest eigenvalues). The number of eigenvectors chosen determines the dimensionality of 'face space', and new images can be classified by a projection onto that face space.

4.6.1.1 Eigenfaces

In mathematical terms, this is equivalent to finding the principal components of the distribution of faces, or the eigenvectors of the covariance matrix of the set of face

images, treating an image as a point (or vector) in a very high dimensional space. The eigenvectors are ordered, each one accounting for a different amount of the variation among the face images.

These eigenvectors can be thought of as a set of features which together characterize the variation among face images. Each image contributes some amount to each eigenvector, so that each eigenvector formed from an ensemble of face images appears as a sort of ghostly face image, referred to as an *eigenface*. Examples of these faces are shown in Figure 4-3. Each individual face image can be represented exactly in terms of a linear combination of the eigenfaces. Each face can also be approximated using only the "best" eigenfaces - those that have the largest eigenvalues, and which therefore account for the most variation within the set of face images. The best M eigenfaces span an M dimensional subspace - "face space" - of the space of all possible images.

Because eigenfaces will be an orthonormal vector set, the projection of a face image into "face space" is analogous to the well-known Fourier transform (FT). In the FT, an image or signal is projected onto an orthonormal basis set of sinusoids at varying frequencies and phase, as depicted in Figure 4-3(a). Each location of the transformed signal represents the projection onto a particular sinusoid. The original signal or image can be reconstructed exactly by a linear combination of the basis set of signals, weighted by the corresponding component of the transformed signal. If the components of the transform are modified, the reconstruction will be approximate and will correspond to linearly filtering of the original signal.

Figure 4-3(b) shows the analogy to the "Eigenface transform". This transform is non-invertible, in the sense that the basis set is small and can reconstruct only a limited range of images. The transformation will be adequate for recognition to the degree that the "face space" spanned by the eigenfaces can account for a sufficient range of faces.

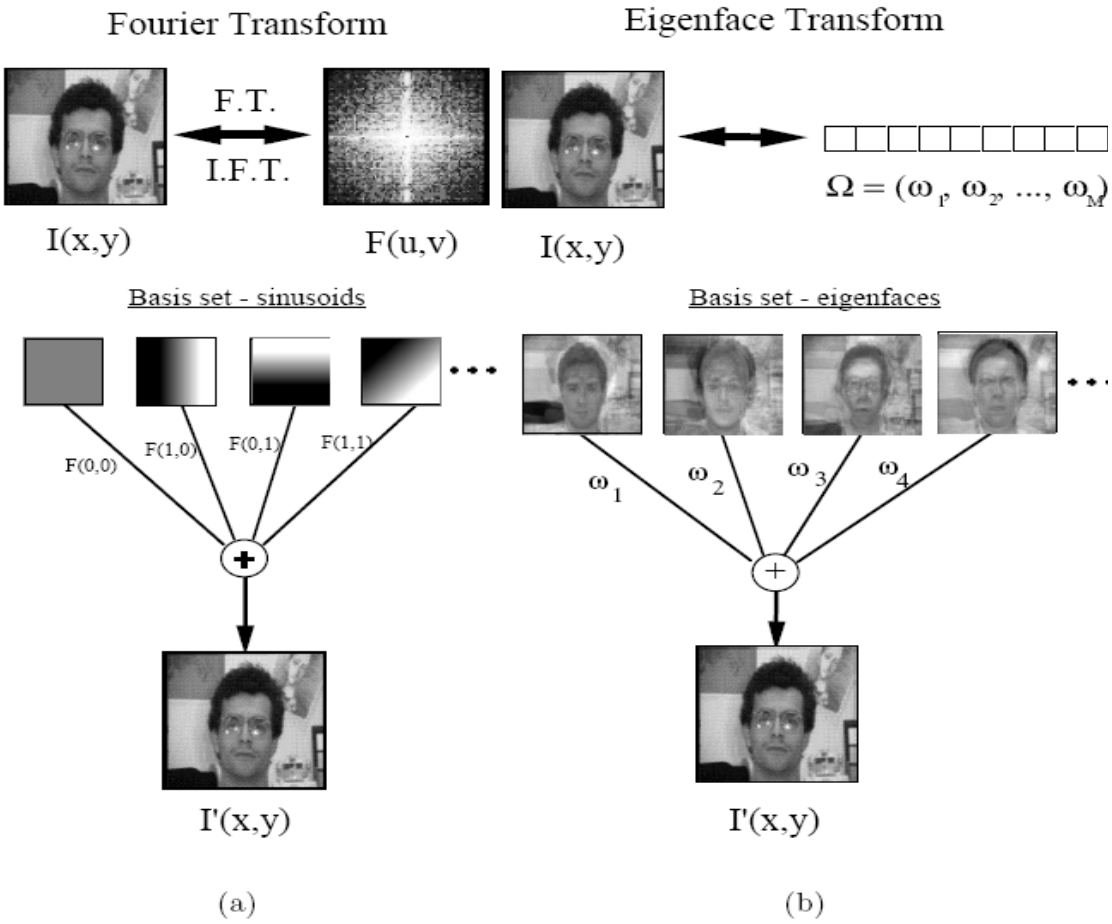


Figure 4-3**:** Transformation and reconstruction of images with [36] (a) the Fourier transform, and (b) the Eigenface transform.

The idea of using eigenfaces was partially motivated by the work of Sirovich and Kirby [25, 27] for efficiently representing pictures of faces using PCA starting with an ensemble of original face images, they calculated a best coordinate system for image compression, where each coordinate is actually an image which they termed an *eigenpicture*. They argued that, at least in principle, any collection of face images can be approximately reconstructed by storing a small collection of weights for each face and a small set of standard pictures (the eigenpictures). The weights describing each face are found by projecting the face image onto each eigenpicture.

4.6.2 Receptive Field-based Approach

The receptive field (RF) of a visual neuron is the area of the visual field (image) where the stimulus can influence its response. For the different classes of these neurons, a receptive field function f(x, y) can be defined. Precomputed filters can simulate such fields when applied to locations across the image. This type of preprocessing is more biologically motivated than simple edge detectors or intensity normalization, as there is psychophysical and physiological evidence for orientation and spatial frequency specific channels in biological visual systems [30].

4.6.2.1 Gabor Filters

The receptive fields of the simple cells in the primary visual cortex (V1) of mammals are oriented and have characteristic spatial frequencies. [30] proposed that these could be modeled as complex 2-D Gabor filters, which have been found to be efficient in reducing image redundancy and robust to noise. Such filters can be either convolved or applied to a limited range of positions, such as for 'jets', where a region around a pixel is described by the responses of a set of Gabor filters of different frequencies and orientations, all centered on that pixel position.

2-D Gabor filter is a product of an elliptical Gaussian in any rotation and a complex exponential representing a sinusoidal plane wave. The sharpness of the filter is controlled on major and minor axis by γ and η. The filter response can be normalized to have a compact closed form [31]

$$\begin{aligned} \psi(x, y; f_0, \theta) &= \frac{f_0^2}{\pi\gamma\eta} e^{-\frac{f_0^2}{\gamma^2}x'^2 + \frac{f_0^2}{\eta^2}y'^2} e^{j2\pi f_0 x'} \\ x' &= x\cos\theta + y\sin\theta \\ y' &= -x\sin\theta + y\cos\theta \end{aligned} \tag{4.1}$$

Where f_0 is the central frequency of the filter, θ is the rotation angle of both the Gaussian major axis and the plane wave, γ is the sharpness along the major axis and η is the sharpness along the minor axis (perpendicular to the wave). The aspect ratio of the Gaussian is λ = η / γ.

The normalized Gabor filter in the frequency domain is

$$\begin{aligned} \Psi(u, v; f_0, \theta) &= e^{-\pi^2 \left(\frac{u'-f_0}{\alpha^2}+\frac{v'}{\beta^2}\right)} \\ u' &= u\cos\theta + v\sin\theta \\ v' &= -u\sin\theta + v\cos\theta. \end{aligned} \tag{4.2}$$

Examples of Gabor filters in 2-D are presented in Figure. 4-4

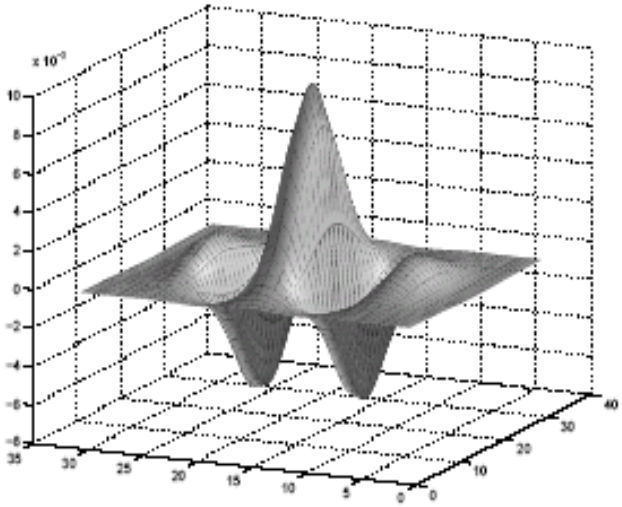


Figure 4-4: Gabor filter in the spatial domain

The form in Eq. (4.1) is centered to the origin and a filter response for an image function $\xi(x, y)$ can be calculated at any location (x, y) with the convolution

$$\begin{aligned} r_\xi(x, y; f, \theta) &= \psi(x, y; f, \theta) * \xi(x, y) \\ &= \iint_{-\infty}^{\infty} \psi(x - x_\tau, y - y_\tau; f, \theta)\xi(x_\tau, y_\tau)dx_\tau dy_\tau \end{aligned} \tag{4.3}$$

Typical Gabor feature, such as Simple Gabor feature space, consists of responses calculated with Gabor filters at several different orientations and scales (frequencies): a filter bank. Using many different orientations and scales ensures invariance; objects can be recognized at various different orientations, scales and translations. Invariance properties of Gabor filters have been presented in more detail in [31, 32].

It can be stated for an image ξ', which equals ξ rotated by Φ, scaled by *a* and intensity multiplied by *c*, that

$$r_{\xi'}(x_0, y_0; f, \theta) = c\, r_{\xi}(a x_0, a y_0; \frac{f}{a}, \theta - \phi) \tag{4.4}$$

That is, geometrical transformations of an object can be captured by filter manipulation. [45]

A filter bank consisting of several filters needs to be used because relationship between responses provides the basis for distinguishing objects. The selection of discrete rotation angles θ_l has already been demonstrated in [34], where it was shown that the orientations must be spaced uniformly.

$$\theta_l = \frac{l2\pi}{n} \;\; l = \{0, \ldots, n-1\} \;\; , \tag{4.5}$$

Where θ_l is the *l*th orientation and *n* is the total number of orientations to be used.
The computation can be reduced to half since responses on angles [π, 2π] are complex conjugates of responses on [0, π] in a case of a real valued input. For the result in Eq. (4.4) to hold the frequencies must be drawn from [31, 33],

$$f_l = k^{-l} f_{max} \;\; l = \{0, \ldots, m-1\}. \tag{4.6}$$

Useful values for *k* include *k* = 2 for octave spacing and *k* = 2 for half-octave spacing.
Now, using the features in Eq. (4.3) and the parameter selection schemes to cover frequencies of interest $f_0, \ldots, fm-1$ and the orientations for desired angular discrimination, one can construct a set of features at an image location (*x*0, *y*0). For instance in simple Gabor feature space a feature matrix G is used [32]

$$\mathbf{G} = \begin{pmatrix} r(x_0, y_0; f_0, \theta_0) & \cdots & r(x_0, y_0; f_0, \theta_{n-1}) \\ r(x_0, y_0; f_1, \theta_0) & \cdots & r(x_0, y_0; f_1, \theta_{n-1}) \\ \vdots & \vdots & \vdots \\ r(x_0, y_0; f_{m-1}, \theta_0) & \cdots & r(x_0, y_0; f_{m-1}, \theta_{n-1}) \end{pmatrix} . \tag{4.7}$$

The feature matrix can be used as an input feature for any classifier. If features are extracted from objects in a standard pose, it is possible to introduce matrix manipulations that allow invariant search [31, 32]. Column-wise circular shift of the feature matrix provides orientation manipulation, though if responses are calculated for only half orientation space the phase wrapping must be taken into consideration. Similarly a row-wise shift provides scale manipulation. In the case of scale manipulation the shift is not circular but the highest frequencies vanish and new lower frequencies are mapped into feature matrix.

4.7 Face Reasoning

Once a database has been collected and a representation decided upon for the images, the method of comparison between exemplar and test faces has to be determined. This reasoning can be simple matching if the representation extracted is extremely face specific or can be very adaptive if a more generalized representation (not very discriminable) is chosen. It can be seen that the type of representation has determined a ‘face-space’ in which distance comparisons can be made. Standard distance metrics, such as Euclidean can be used for matching, whereas simple weighted sums may be more suitable for internal ‘hidden’ representations.

Learning is an important factor in any useful application, to avoid the ’brittleness’ commonly found in manually extracted rule systems. Even simple vision tasks are of such complexity that original assumptions in manual systems turn out not to be valid or only partially valid in certain circumstances. In addition, such an approach is neither scalable nor modifiable in day to day operation. For example, if the task changes from the original specification due to different people or rooms being involved, the system should be able to automatically relearn the task, rather than require an operator to reprogram new rules to cover the changed circumstances.

4.7.1 Connectionist Approach

Neural networks have a long history of being used for FR, though computational limitations of the time seem to have restricted the amount of testing that was possible.

The Kohonen associative networks were able to demonstrate quite early, on one of the main advantages of the distributed processing in neutral networks, which is a tolerance to noisy or incomplete test data. They could classify grey-level images of faces when a forcing stimulus (the desired output activity) was provided along with the stimulus pattern (the input data). These values were clamped until a steady state of activations was reached. The idea was that, when unclamped, the network would converge when given the original input to give the desired output values. It could also generalize in classifying new views of learnt faces by interpolating within the range of angles already seen, but could not extrapolate to images outside this area.

MLP, commonly trained using gradient descent with error back-propagation, is capable of good generalization for difficult problems, but is notoriously difficult to ensure global convergence under all training runs, as the non-linearity of the hidden units and the nature of the input-output mapping lead to a large number of local minima, and training times can typically be long. [28, 29] used multi-layer networks with target output equal to input (auto-association) in order to compress photographic images. The network was trained on random patches of image. The compressed signal could be taken from the hidden layer of units (these values were effectively eigenvalues, the eigenvectors, called ‘holons’ here, being contained in the weight values between the unit layers), and these values could, in turn, be put back in to decode or uncompress the original image as output values.

Much of the literature on face recognition with neural networks presents results with only a small number of classes (often below 20).

In [40] the first 50 principal components of the images are extracted and reduced to 5 dimensions using an autoassociative neural network. The resulting representation is classified using a standard multi-layer perceptron. Good results are reported but the database is quite simple: the pictures are manually aligned and there is no lighting

variation, rotation, or tilting. There are 20 people in the database. A hierarchical neural network which is grown automatically and not trained with gradient-descent was used for face recognition by Weng and Huang [42]. They report good results for discrimination of ten distinctive subjects.

In [39] a HMM-based approach is used for classification of the ORL database images. The best model resulted in a 13% error rate. Samaria also performed extensive tests using the popular eigenfaces algorithm [36] on the ORL database and reported a best error rate of around 10% when the number of eigenfaces was between 175 and 199. In [41] Samaria extends the top-down HMM of [39] with pseudo two-dimensional HMMs. The error rate reduces to 5% at the expense of high computational complexity – a single classification takes four minutes on a Sun Sparc II. Samaria notes that although an increased recognition rate was achieved the segmentation obtained with the pseudo two-dimensional HMMs appeared quite erratic.

CHAPTER 5 IMPLEMENTATION OF FEATURE EXTRACTION TECHNIQUES

5.1 System Components

The system proposed for FR has the following main and important components. A high-level block diagram is shown in figure 5-1.

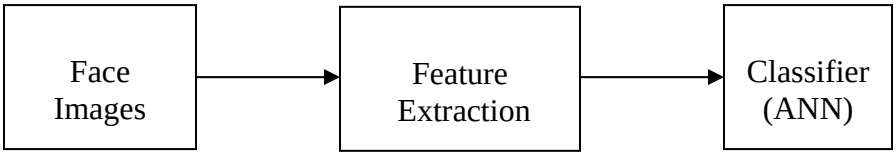


Figure 5-1: A high level block diagram of the proposed face recognition system

5.2 Utilized Face Database

5.2.1 Olivetti Research Laboratory (ORL) Face Database

The database used is the ORL database which contains photographs of faces taken between April 1992 and April 1994 at the ORL in Cambridge, UK. There are 10 different images of 40 distinct subjects as shown in figure 5-2. For some of the subjects, the images were taken at different times. There are variations in facial expressions (open/closed eyes, smiling/non-smiling) and facial details (glasses/no glasses). All of the images were taken against a dark homogeneous background with the subjects in an upright, frontal position, with tolerance for some tilting and rotation of up to about 20 degrees. There is some variation in scale of up to about 10%. The images are grayscale with a resolution of 92 * 112. The variations in the images corresponding to subject number 40 are shown in figure 5-3. Most FRS are developed and tested using this ORL, the standard face database. As there are 10 images per individual in the ORL database, 6 of them are used to train the system and the rest are used to test it.

Figure 5-2: The 40 distinct subjects in ORL

Figure 5-3: The set of 10 images of the 40th subject. Considerable variations are seen

5.3 Feature Extraction Techniques

Two distinct models are developed to extract the features from the face images. The first model uses PCA and is explained in detail in section 5.3.1 and the second model uses GF and PCA and is explained in section 5.3.2.

5.3.1 Principal Component Analysis

This transform is designed in such a way that the dataset may be represented by a reduced number of "effective" features and yet retains most of the intrinsic information of the data; in other words, the data set undergoes a dimensionality reduction.

The modules used in this extraction technique are shown in Figure 5 -4.

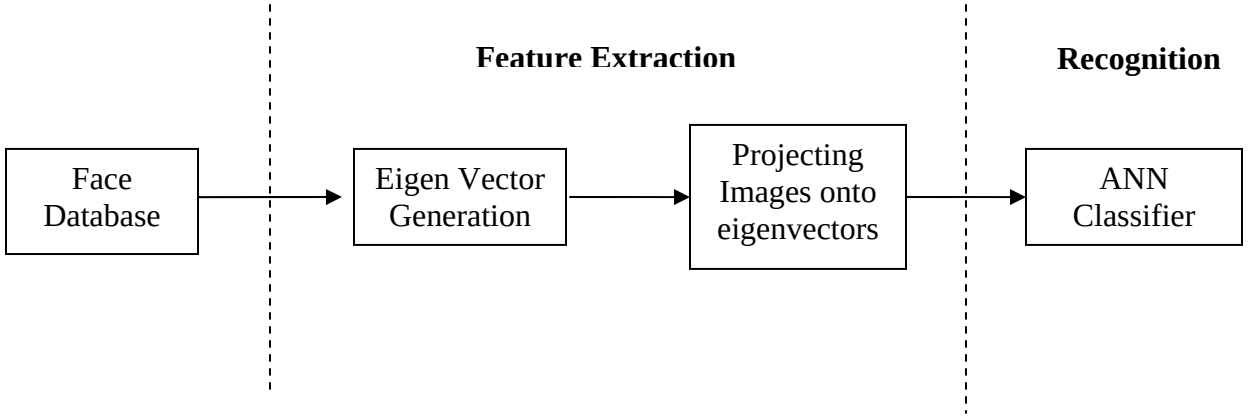
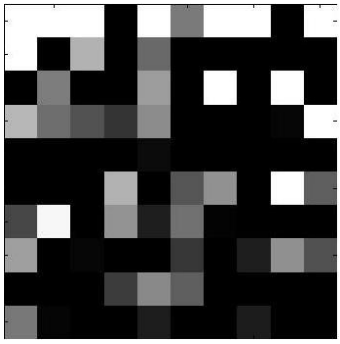


Figure 5-4: Proposed system setup one

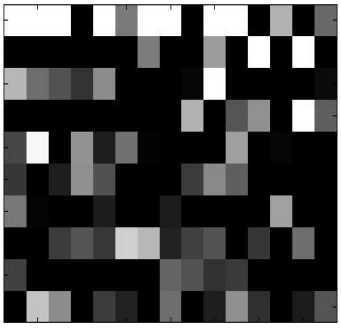
While doing PCA, (described in Appendix C), one has to clarify regarding the number of training patterns available to PCA and the dimensionality reduction required from PCA. ORL face database has 400 images and each image has 10304 (112*92) pixels. The number of pixels defines the number of dimensions. Next step is to find the covariance matrix and its eigenvectors and eigenvalues. Reorder eigenvectors so as to be in a descending order of "importance" (first eigenvectors are those whose corresponding eigenvalues are the maximum). Then depending on the dimensionality reduction level one wants to achieve, appropriate number of less important eigenvectors can be eliminated.

a) Original Image



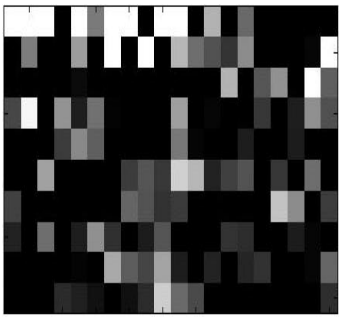
b) First 100 Eigen vectors

c) reconstructed original image



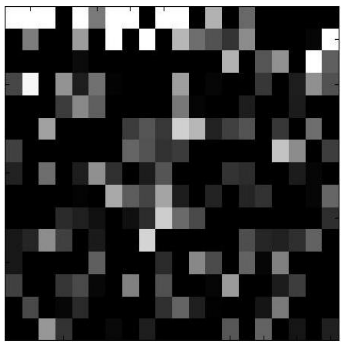
d) First 150 Eigen vectors

e) reconstructed original image



f) First 200 Eigen vectors

g) reconstructed original image



h) First 300 Eigen vectors

i) reconstructed original image

Figure 5-5: First 100, 150, 200 and 300 principal components from a ORL dataset composed of 112 * 92 pixel images and the four corresponding projections of the principal components to the initial 10304-dimentional space.

Number of Eigenvectors	Reconstruction MSE
100	13.449
150	11.082
200	9.0277
300	4.7985

Table 5-1 Mean Square Error (MSE) when only a subset of principal components are used to reconstruct the original image

In this thesis, the first 100, 150, 200 and 300 principal components (eigenvectors) of the covariance matrix are investigated (see Figure 5-5). The first 200 eigenvectors are selected for further work as it provides sufficient dimensionality reduction and optimal MSE when it is used to reconstruct the original signal (see Table 5-1). Though the MSE for 300 eigenvectors is very small, it is not selected for further work. This is due to the fact that the increase in vector size will intensify the computations and increase the time for developing an optimal ANN model. So, there is a trade-off between the feature vector size and the computations. Hence, in this work, average eigenvector of size 200 is used for training and testing the ANN model.

5.3.2 Gabor Feature Representation

PCA is an efficient way of reducing dimensionality, but has the drawback of being more sensitive to image variations. Its performance is dependent on the accuracy of normalization, and the process has no inherent invariance to translation, scale or rotation [38]. So, Gabor wavelet transform have been performed before PCA [37] to provide a greater level of invariance than found using grey-level pixel information. Gabor coefficients can be used as data for PCA to provide a greater level of illumination invariance than found using grey-level pixel information [37].

The Gabor wavelet representation of an image is the convolution of the image with a family of Gabor kernels as defined by Eq. 4.1. To facilitate the Gabor filter representation, the ORL images are scaled to 128 X 128 using a bicubic interpolation. The high level block diagram for feature extraction involving GF is shown in figure 5-6.

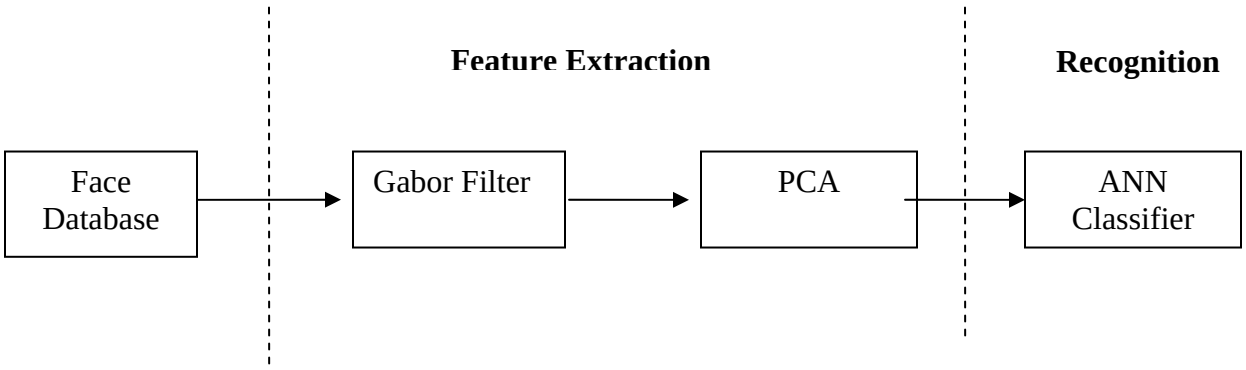


Figure 5-6: Proposed System setup two

Let *I(x, y)* be the gray level distribution of an image *I*, the convolution output of image and a Gabor kernel $\psi_{\mu,v}$ is defined as follows:

$$O_{\mu,\nu}(z) = I(z) * \psi_{\mu,\nu}(z) \qquad 5\text{-}1$$

Where z = (x, y), and * denotes the convolution operator.

Applying the convolution theorem, we can derive the convolution output from Eq. 5-1 via the fast Fourier transform (FFT)

$$\mathfrak{F}\{O_{\mu,\nu}(z)\} = \mathfrak{F}\{I(z)\}\mathfrak{F}\{\psi_{\mu,\nu}(z)\} \quad 5.2$$

And

$$O_{\mu,\nu}(z) = \mathfrak{F}^{-1}\{\mathfrak{F}\{I(z)\}\mathfrak{F}\{\psi_{\mu,\nu}(z)\}\} \quad 5.3$$

Where $\Im$ and $\Im^{-1}$ denote the Fourier and inverse Fourier transform, respectively

The convolution outputs (both the real part and the magnitude) of a sample image (the first image in Figure 5-8) and those Gabor kernels (Figure 5-7) are shown in Figure 5-9. The outputs exhibit strong characteristics of spatial locality, scale, and orientation selectivity corresponding to those displayed by the Gabor wavelets in Figure 5-7. Such characteristics produce salient local features, such as the eyes, nose and mouth that are suitable for visual event recognition.

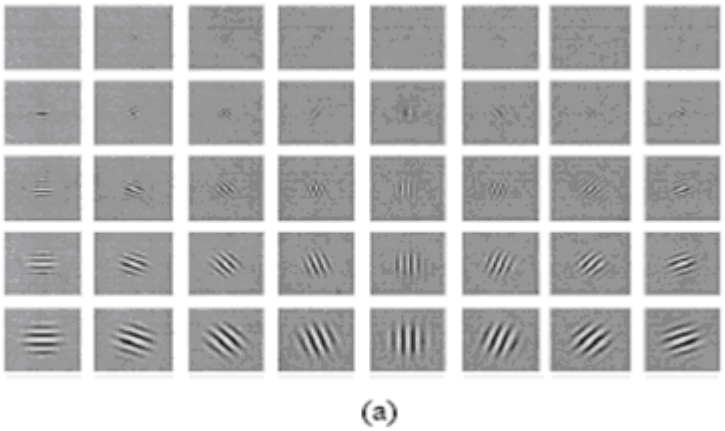


Figure 5-7: Gabor wavelets (a) Real part of the Gabor kernels at five scales and eight orientations for $\lambda=2\pi$, $f_{max} = \pi / 2$, and $k=\sqrt{2}$

Figure 5-8: Example of ORL images used in our experiments

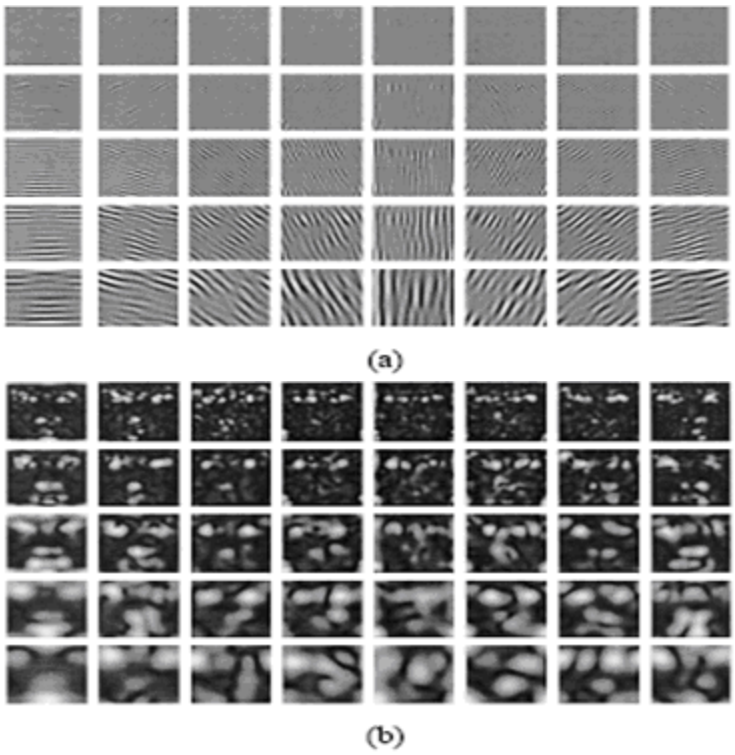


Figure 5-9: Convolution outputs of a sample image (the first image in Figure 5-8) and the Gabor kernels (Figure 5-7). (a) Real part of the convolution outputs. (b) Magnitude of the convolution outputs.

Since the outputs $O_{\mu,\nu}(z)$ ($\mu \in \{0,\ldots,7\}$,$\nu \in \{0,\ldots,4\}$) consist of different local, scale, and orientation features, we concatenate all these features in order to derive a feature vector χ. As a result, the feature vector consists of both the real and the imaginary part of the Gabor transform and its length is 655360. Without loss of generality, each output is assumed to be a column vector, which can be constructed by concatenating the rows (or columns) of the output. Before the concatenation, each output is first down sampled by a factor ρ=64 to reduce the dimensionality of the original vector space. Let $O^{(\rho)}_{\mu,\nu}$ denote a normalized output (downsampled by 64), then the feature vector has a length of 10240 and is defined as follows:

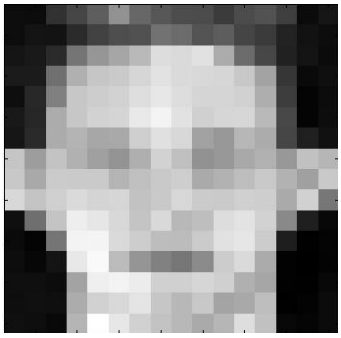
$$X^{(\rho)} = \left(O_{0,0}^{(\rho)t} \quad O_{0,1}^{(\rho)t} \quad \cdots \quad O_{4,7}^{(\rho)t} \right)^t \qquad 5.4$$

Where, t is the transpose operator. The feature vector thus encompasses all the outputs, $O_{\mu,\nu}(z)$ ($\mu \in \{0,\ldots,7\}$,$\nu \in \{0,\ldots,4\}$), as important discriminating information.

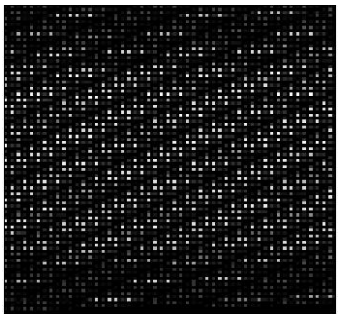
Figure 5-10 illustrates the output of GF and PCA techniques and the MSE while trying to reconstruct the GF image for various eigenvector lengths. 200 eigenvectors of the Covariance matrix formed from Gabor feature vectors are chosen (as explained in the previous section) as those with the largest associated eigenvalues. A face space is constructed using these 200 eigenfaces and each Gabor feature vector is projected to it. The dimension of the Gabor feature vector (10240) is thus reduced to 200 and it can be used as a feature vector to train or test ANN.

a) Original Image

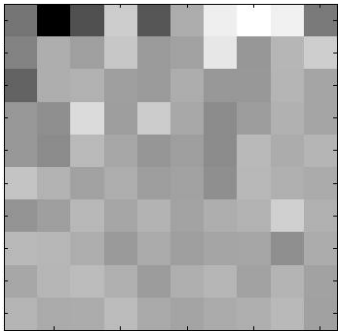
b) output of Gabor Filter



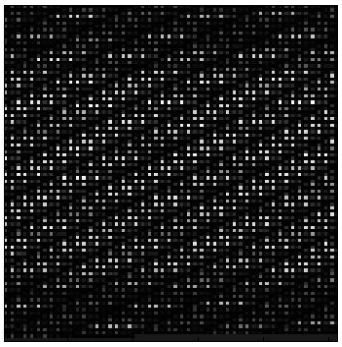
c) down sampled Image



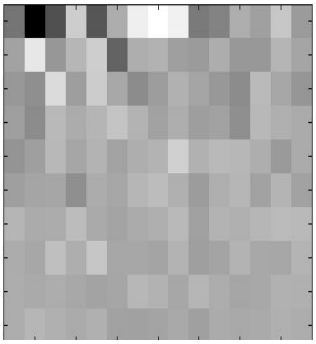
d) Input to PCA module



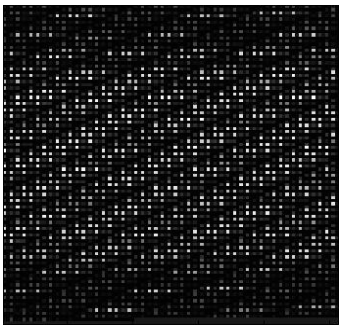
e) First 100 Eigen vectors



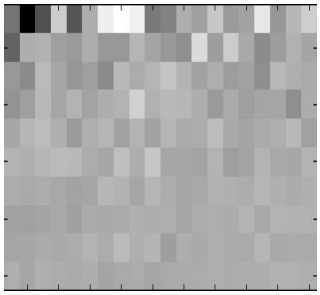
f) reconstructed image



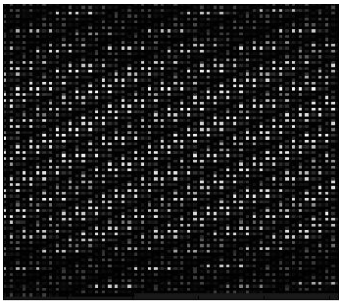
g) First 150 Eigen vectors



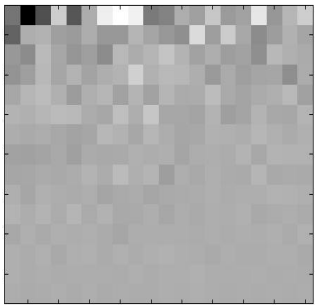
h) reconstructed image



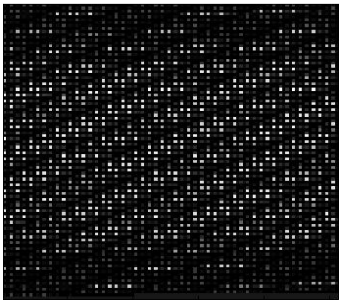
i) First 200 Eigen vectors



j) reconstructed image



k) First 300 Eigen vectors



l) reconstructed image

Figure 5-10: Output of the GF and PCA modules for different number of Eigen vectors

Number of Eigenvectors	Reconstruction MSE
100	0.055052
150	0.032981
200	0.018768
300	0.0044954

Table 5-2 MSE when only a subset of principal components are used to reconstruct Gabor Filtered image

CHAPTER 6 RECOGNITION RESULTS

6.1 Introduction

The output vector from the feature extraction technique is given as input to the ANN model for the classification purpose. The purpose of the model is to identify the test image against a face database.

6.2 Neural Network classifier

This chapter discusses in detail some of the decisions taken on several issues that arose during the construction of ANN classifier for the task of FR. It is the final and most important step, though the success of such a recognition system does not exclusively rely on ANN optimization. All the three steps (Acquisition, Feature Extraction, and Classifier) for the development of a recognition system need care. What kind of weight initialization is used and why, how the outputs are interpreted, how the dataset is divided for training and testing, how the network parameters and network topology influence the performance of the network are some of the issues that are illuminated in this chapter.

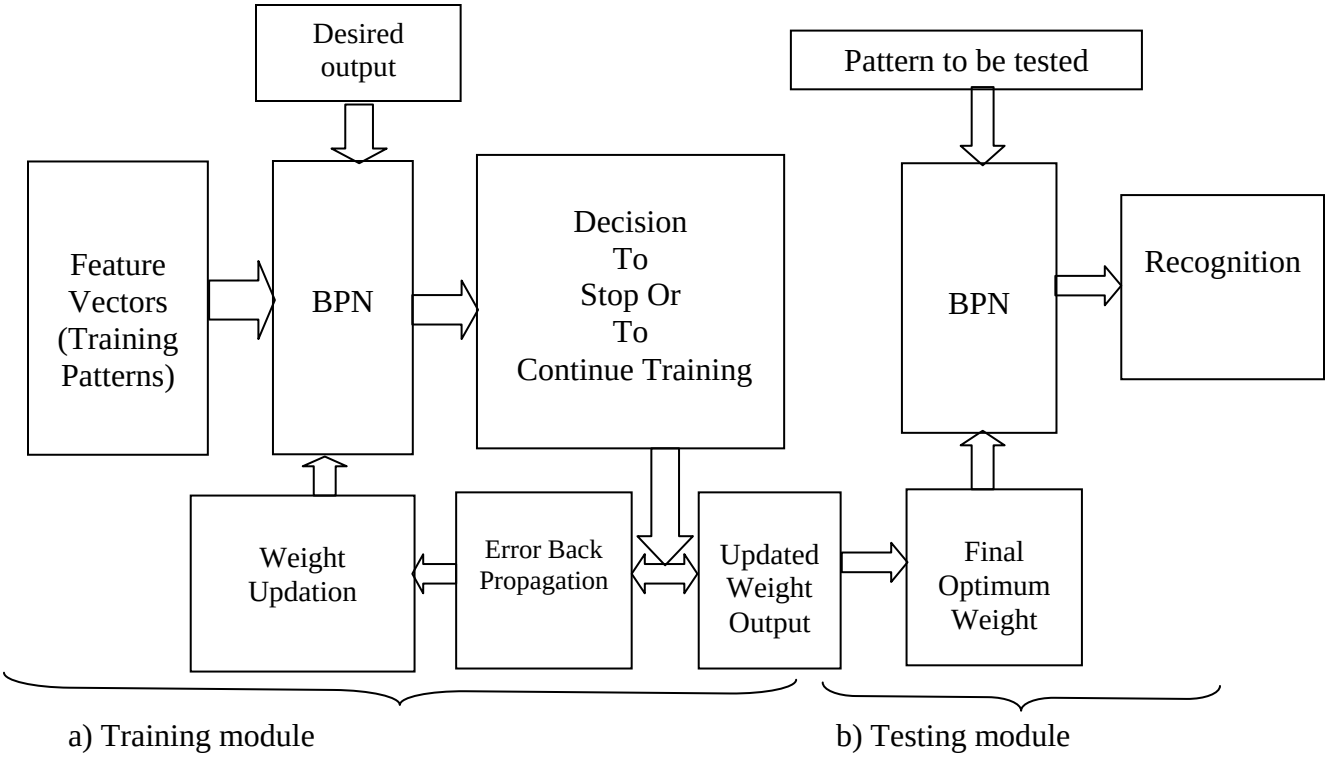


Figure 6-1: Functional Block Diagram of Artificial Neural Network Classifier

6.2.1 Network Architecture and Parameters

MLP which is one of the popular ANN model is used in this thesis work. MLP is trained using BPA. MLP is a three layer feed forward network to be trained in supervised learning mode. The first layer of the network is the input layer, which is used to distribute the training patterns uniformly to the subsequent layers and since the number of nodes in this layer must be equal to the number of input data points per pattern, it is set to 200. The number 200 was selected based on the length of the feature vector of the training and testing patterns.

The second and third layers are the hidden layer and the output layer respectively which play the role of classification or generating decision boundaries used in the categorization of the training patterns into their true classes via the adaptive adjustment and convergence of the input-to-hidden layers and hidden-to-output layers weighting coefficients to an acceptable value determined by the output layer sum of squared error value at the last iteration.

The number of nodes in the hidden layer is determined in conjunction to the number of input layer nodes, output layer nodes and other network parameters together with the evaluation of the compromise between learning performance (convergence speed) and test performance (acceptance of the solution arrived). Based on these facts, it is set to 100.

The output layer nodes are also used to monitor the learning performance of the network at each iteration. The number of nodes "*M*" required in this layer is determined by the number of subjects "*N*" to be classified. It can be shown that those two parameters do have logarithmic relationships so that the number of nodes at the output layer can be found by using the formula $M = ceil(\log^{N}{}_{2})$. So, in the present case for N = 40, M = 6. The final outputs are passed through a thresholding function of value 0.5. The output of the neuron will be equal to '1' if the activation value is ≥ 0.5, else it will be equal to '0'. The value is selected experimentally and the same is explained in section 6.8 of this chapter.

Initial input-to-hidden and hidden-to-output layers' weighting coefficients are generated randomly between -0.5 and 0.5, the bias term as usual is set to 1, the learning and momentum rates are set experimentally.

6.2.2 Test, Train and Validation set

The last decision is how to divide the dataset before starting the training and testing. In this system, 60% of the face database is used for training and 40% for testing. The face database consists of 400 images from 40 different subjects, 10 images per subject.

All the 400 images from the ORL database are used in this experiment. Six images are chosen for training and one for validation, on the basis of their intra-subject variation, from the ten images available for each subject, while the remaining three images along with the image used for validation during the training period (unseen during training) are used for testing.

In the proposed system, standard deviation of feature vectors is used to divide the dataset. The feature vectors that represent each subject are sorted in the increasing order of their standard deviation. Then, representative vectors are taken for training, validation and testing (see Figure 6-3).

Figure 6-2: 10 different images for subject number '1'

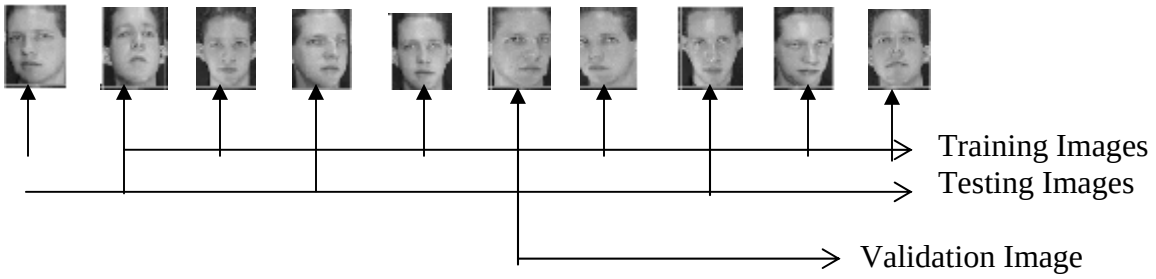


Figure 6-3: Images sorted in the increasing order of their standard deviation

6.2.3 Stopping Criterion

In the proposed system, the stopping criterion of ANN training is determined by the validation set. After every 10 epochs during training, the network is tested with the validation data. The training is stopped as soon as the error on validation set increases more than the previous validation check even if it has not reached the preset epoch number. In the present case, the epoch number is set to 2000. However, the training is stopped whenever the validation error increases. This happened in the case of $\eta = 0.9$ and $\alpha = 0.5$ where the validation error increases after 580^{th} epoch (validation error = 0.14851). So, the training is stopped at that point and the corresponding weight vector is used for the testing purpose. In all the other cases, the minimum validation error occurs at the preset epoch number.

6.2.4 Optimal Learning and Momentum Constants

Using MLP with 100 hidden neurons, different combinations of $\eta \in \{0.01, 0.1, 0.5, \text{and } 0.9\}$ and $\alpha \in \{0, 0.1, 0.5, \text{and } 0.9\}$ are simulated to observe their effect on network convergence. Each combination is trained with the same set of initial random weights and the same set of 240 input-output patterns, so that the results of the experiments may be compared directly.

6.3 Experiment one - Using PCA for Feature Extraction

The performance of the network, with different values of η and α, for the first model is presented in this section. It is clear from the graphs (Figure 6-4 to Figure 6-8) that the learning and validation error decreases over epochs except in two cases where $\alpha = 0.9$. After a certain number of epochs, there is not much change in the error curves. This may be due to the reason that the error point might have either reached the global minimum or local minimum and the network is not learning much later. Also, the tabular results (Table 6.1) demonstrate the effect of η. As the η value increases, the network learning and validation errors also show improvement by large percentage. The network could achieve a good recognition rate with $\alpha = 0.5$ and $\eta = 0.9$.

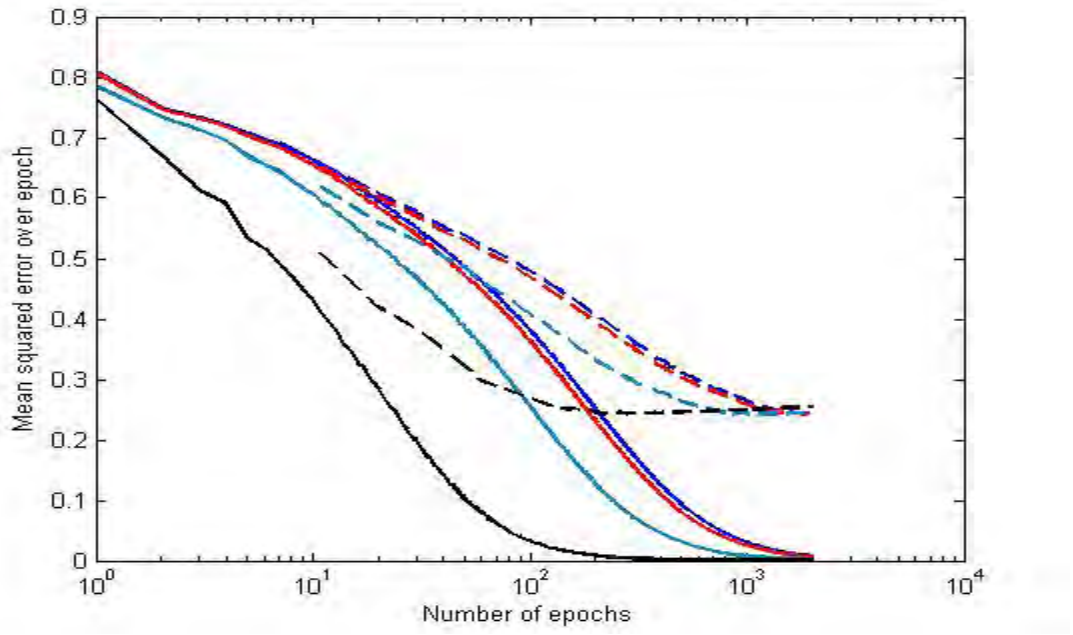


Figure 6-4: Learning and Validation Curves; for varying α and η=0.01 during experiment one

α = 0 α = 0.1 α = 0.5 α = 0.9

Training Validation

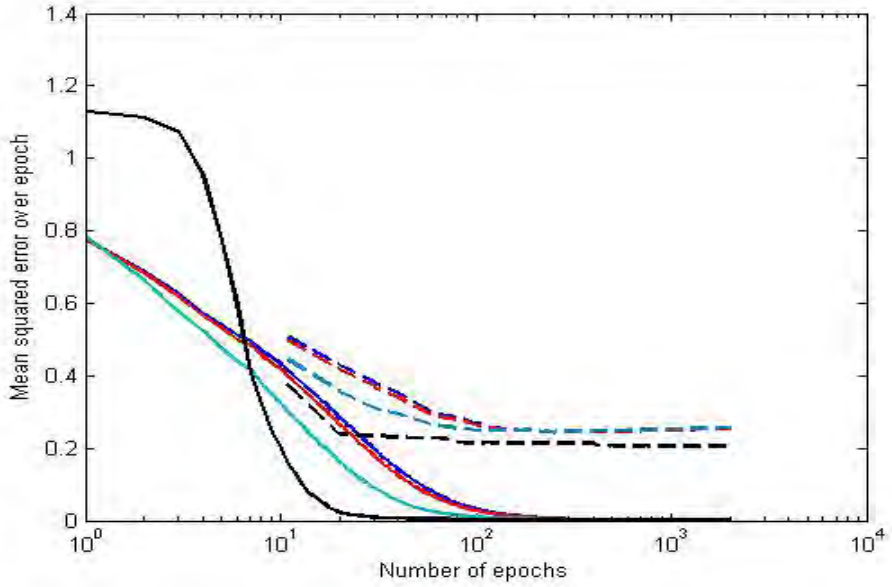


Figure 6-5: Learning and Validation Curves; for varying α and η=0.1 during experiment one

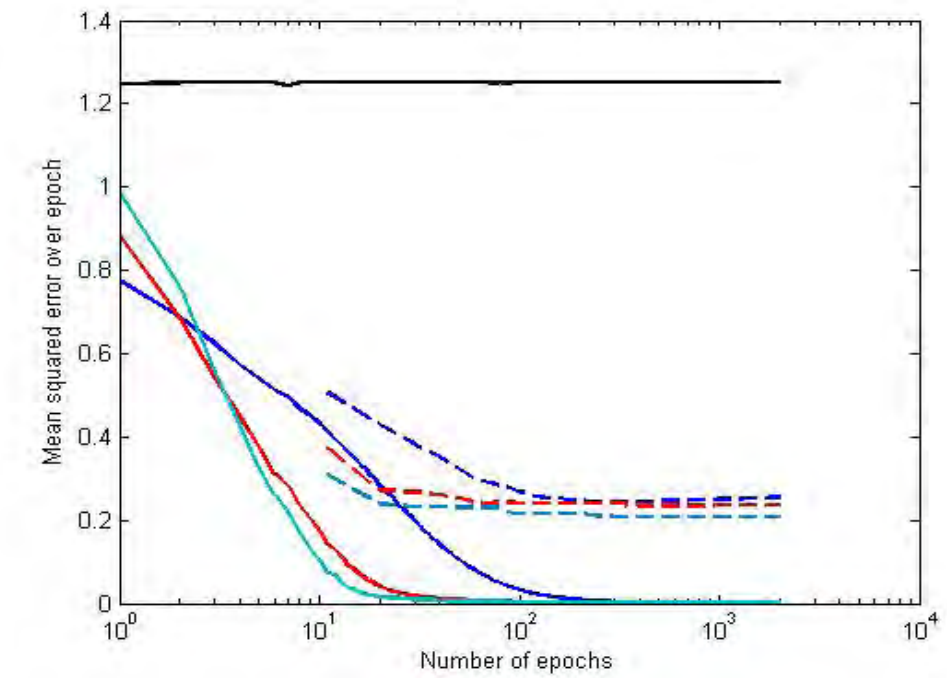


Figure 6-6: Learning and Validation Curves; for varying α and η = 0.5 during experiment one

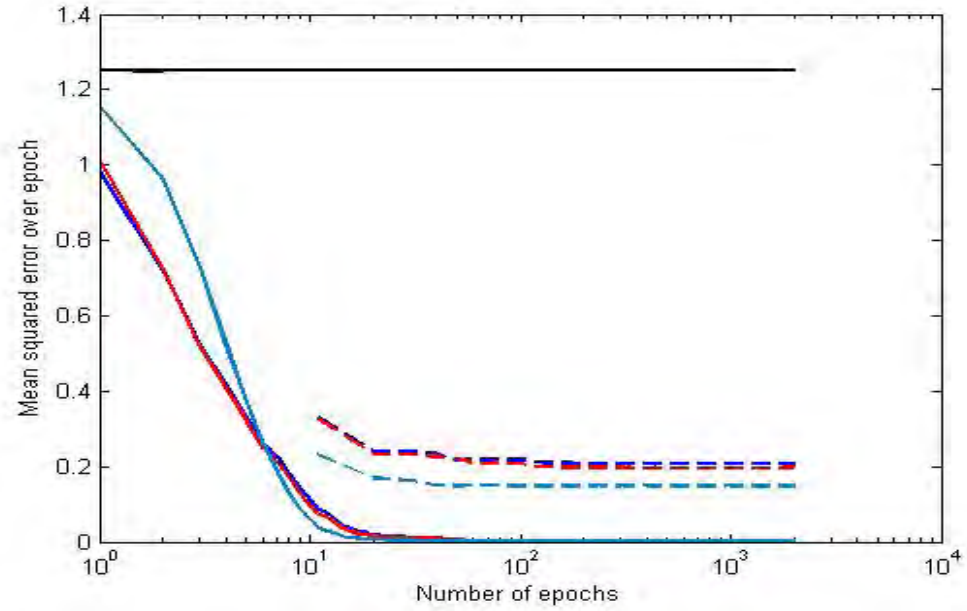


Figure 6-7: Learning and Validation Curves; for varying α and η = 0.9 during experiment one

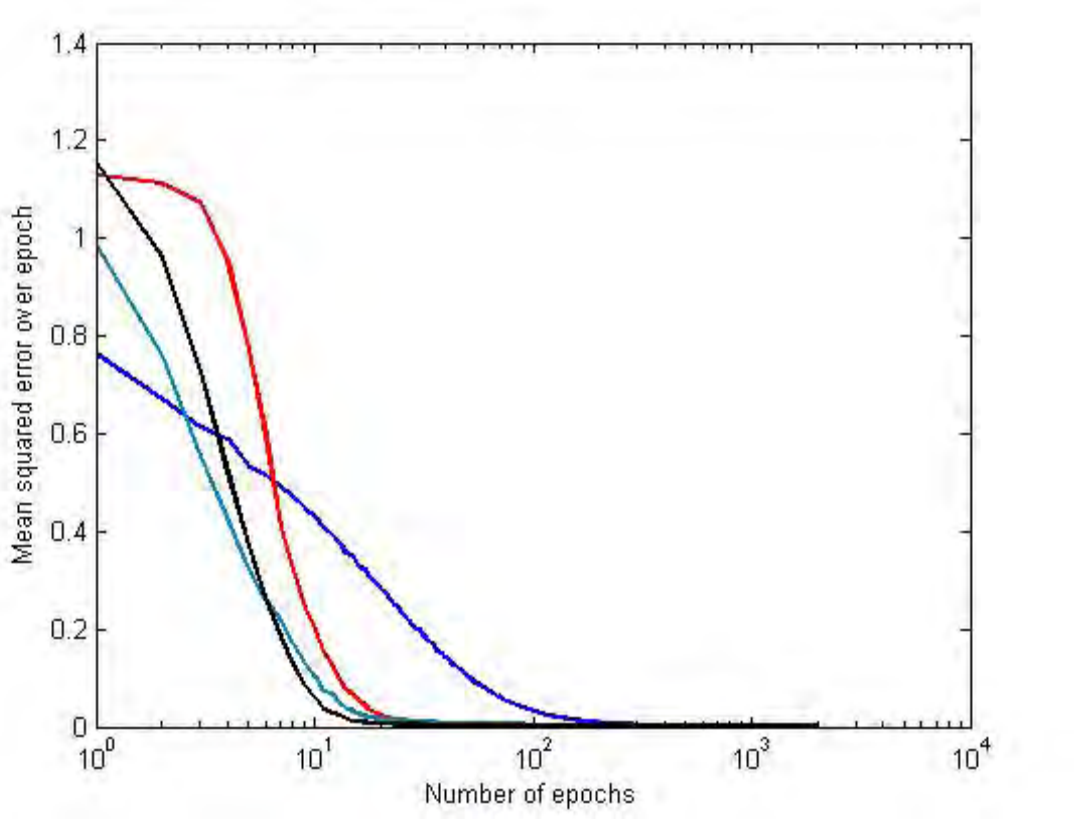


Figure 6-8: Best learning curves of experiment one; selected from Figure 6-4 to 6-7

η = 0.01			
α	**Minimum Training Error**	**Minimum Validation Error**	**Recognition Rate**
0	0.0076408	0.24327	63.33333333
0.1	0.006312	0.24297	62.5
0.5	0.0023899	0.24281	61.66666667
0.9	**0.00026806**	**0.2427**	**63.33333333**

η = 0.1			
α	**Minimum Training Error**	**Minimum Validation Error**	**Recognition Rate**
0	0.00026847	0.24253	63.33333333
0.1	0.00023587	0.24277	63.33333333
0.5	0.00011599	0.24488	63.33333333
0.9	**0.000017513**	**0.20568**	**68.33333333**

η = 0.5			
α	**Minimum Training Error**	**Minimum Validation Error**	**Recognition Rate**
0	0.000038885	0.23646	65
0.1	0.000034389	0.23459	65
0.5	**0.000017343**	**0.2075**	**71.66666667**
0.9	1.2429	1.2498	0

η = 0.9			
α	**Minimum Training Error**	**Minimum Validation Error**	**Recognition Rate**
0	0.000019343	0.20672	69.16666667
0.1	0.000017167	0.19615	70.83333333
0.5	**8.8058E-06**	**0.14851**	**76.66666667**
0.9	1.25	1.25	0

Table 6.1 Training, Validation and Testing Performance of the network for different values of α and η during experiment one

6.4 Experiment two - Feature Extraction using Gabor Filter

Again for the second model with GF and PCA, the performance of the network is analyzed with different values of η and α and the results are presented in this section.

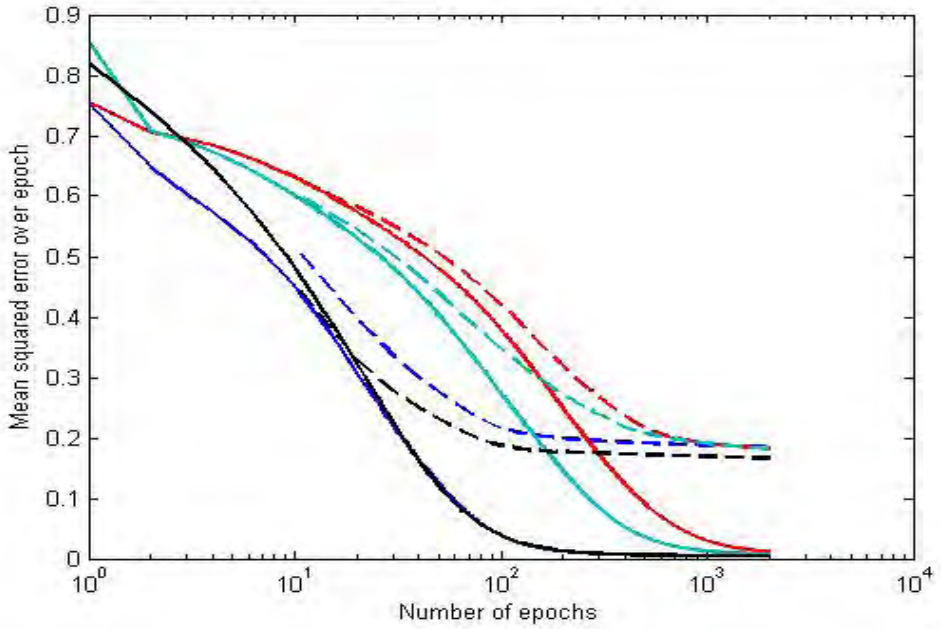


Figure 6-9: Learning and Validation Curves; for varying α and η = 0.01 during experiment two

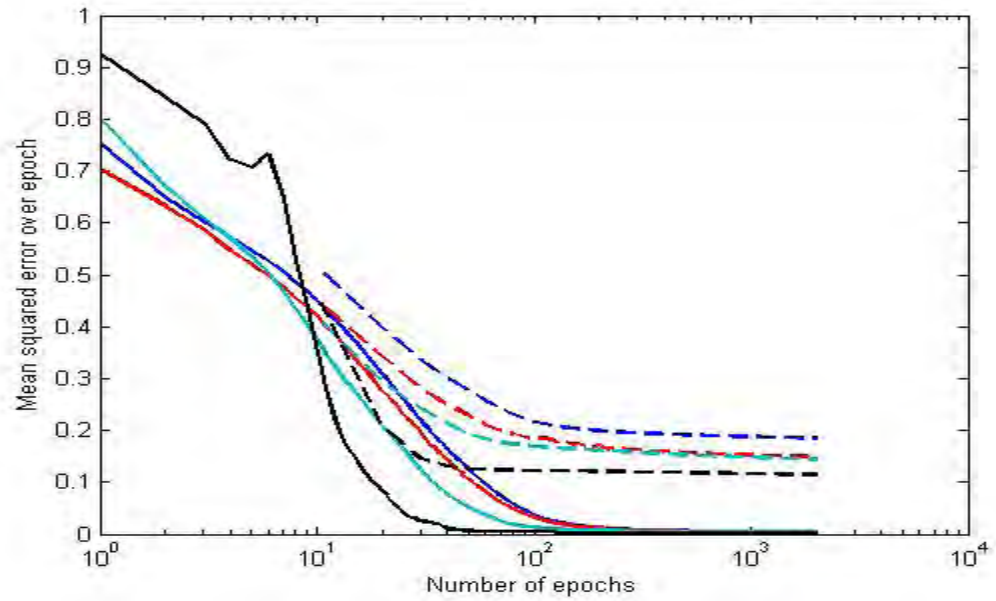


Figure 6-10 Learning and Validation Curves; for varying α and η = 0.1 during experiment two

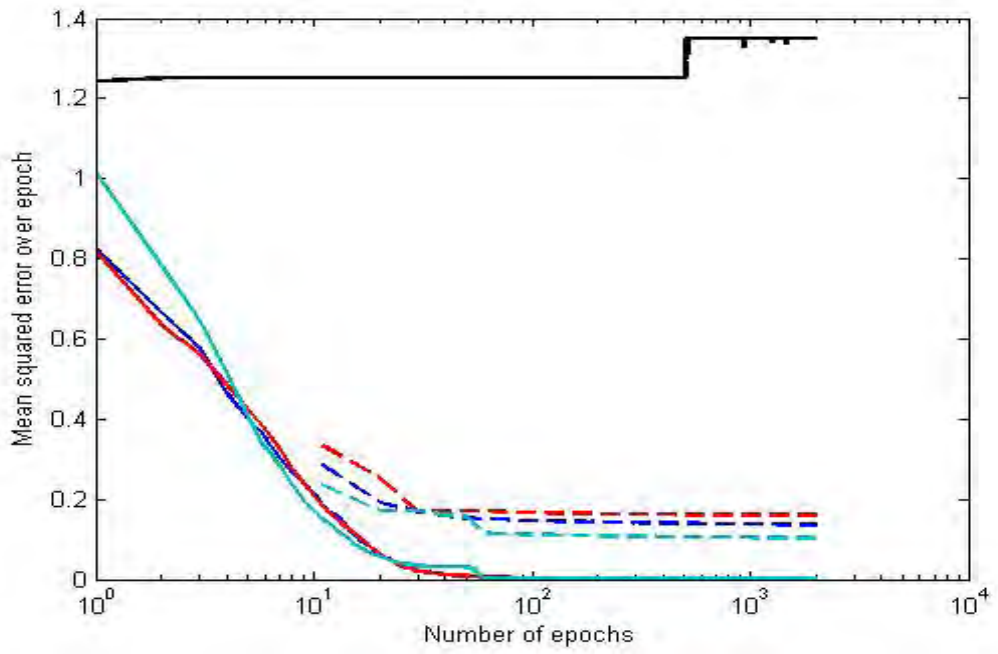


Figure 6-11: Learning and Validation Curves; for varying α and η = 0.5 during experiment two

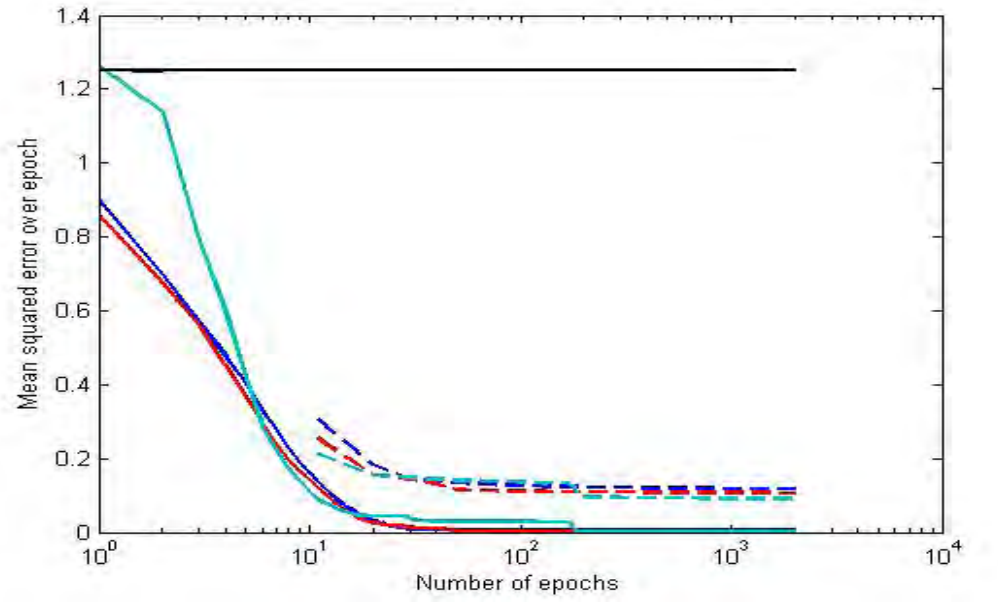


Figure 6-12: Learning and Validation Curves; for varying α and η = 0.9 during experiment two

The experimental learning curves shown here suggest the following trends:

- While, in general, a smaller η results in slower convergence, it can locate "deeper" local minima in the error surface than a larger η. This finding is intuitively satisfying, since a smaller η implies that the search for a minimum should cover more of the error surface than would be the case for a larger η.
- For $\eta \rightarrow 0$, the use of $\alpha \rightarrow 1$ produces increasing speed of convergence. On the other hand, for $\eta \rightarrow 1$, the use of $\alpha \rightarrow 0$ is required to ensure learning stability.
- The use of the constant $\eta = \{0.5, 0.9\}$ and $\alpha = 0.9$ cause oscillations in the mean squared error during learning and a higher value for the final mean-squared error at convergence, both of which are undesirable effects.

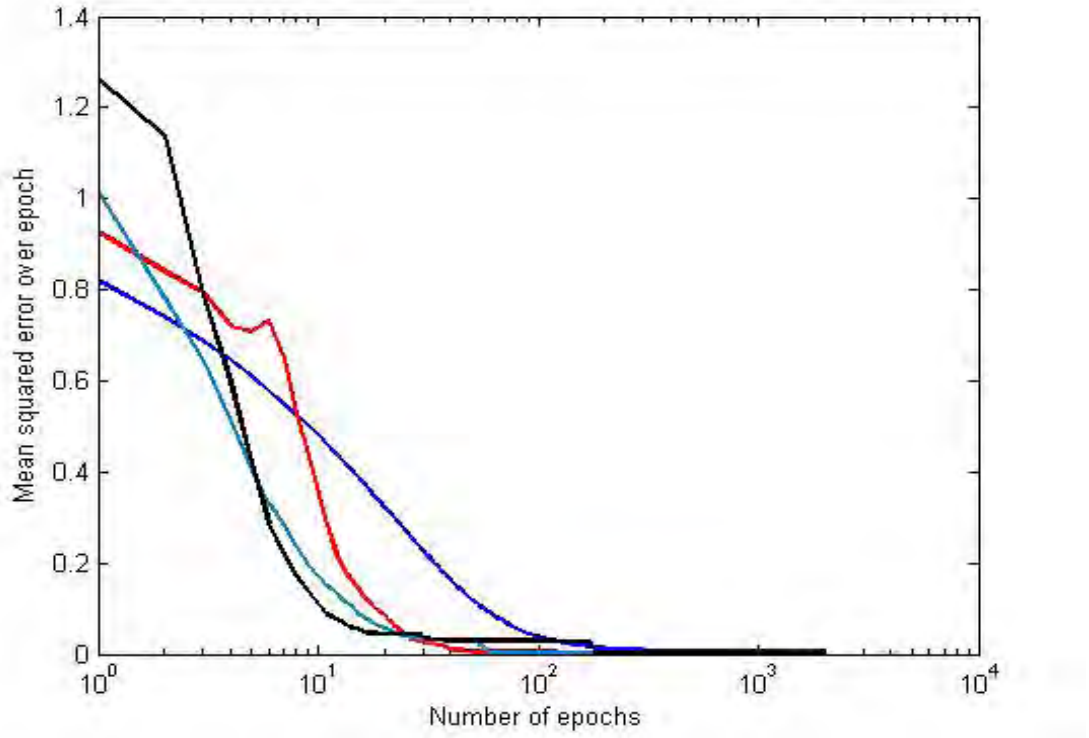


Figure 6-13: Best learning curves of; experiment two selected from Fig. 6-9 to 6-12

$\eta = 0.01, \alpha = 0.9$ $\eta = 0.5, \alpha = 0.5$ Training

$\eta = 0.1, \alpha = 0.9$ $\eta = 0.9, \alpha = 0.5$ Validation

From Fig. 6-13, it appears the optimal learning-rate parameter $\boldsymbol{\eta_{opt}}$ is about **0.9** and the optimal momentum constant $\boldsymbol{\alpha_{opt}}$ is about **0.5**.

η = 0.01			
α	**Minimum Training Error**	**Minimum Validation Error**	**Recognition Rate**
0	0.012518	0.17232	70.8%
0.1	0.011205	0.18245	69.1%
0.5	0.0067885	0.18208	69.1%
0.9	**0.0044579**	**0.16629**	**70.8%**

η = 0.1			
α	**Minimum Training Error**	**Minimum Validation Error**	**Recognition Rate**
0	0.0044568	0.18479	71.6%
0.1	0.0044148	0.1488	72.5%
0.5	0.0042898	0.1442	71.6%
0.9	**0.000021415**	**0.11245**	**77.5%**

η = 0.5			
α	**Minimum Training Error**	**Minimum Validation Error**	**Recognition Rate**
0	0.0044148	0.1488	77.5%
0.1	0.0021196	0.15965	71.6%
0.5	**0.0021015**	**0.1034**	**78.7%**
0.9	1.1246	1.2495	0%

η = 0.9			
α	**Minimum Training Error**	**Minimum Validation Error**	**Recognition Rate**
0	0.0041874	0.11754	79.2%
0.1	0.0021016	0.10629	74.2%
0.5	**9.13E-06**	**0.090955**	**88.3%**
0.9	1.25	1.25	0%

Table 6.2: Training, Validation and Testing Performance of the network for varying α and η during experiment two

6.5 Testing the Network

During testing, test patterns without its associated target are propagated through the trained network. These patterns are weighted (successively by the optimum input-to-hidden layer and hidden-to-output layer weighting coefficients), summed and thresholded at each node of the hidden and output layers. In general, the performance of the proposed system for the optimal (best) parameter values is summarized in Table 6.3. These results correspond to model 2 (GF + PCA + ANN).

Subject Number	Number of Test Images	Recognized Images	Rejected Images	Recognition Rate
1	4	2	2	50%
2	4	4	0	100%
3	4	4	0	100%
4	4	4	0	100%
5	4	3	1	75%
6	4	4	0	100%
7	4	2	2	50%
8	4	3	1	75%
9	4	4	0	100%
10	4	4	0	100%
11	4	4	0	100%
12	4	4	0	100%
13	4	2	2	50%
14	4	3	1	75%
15	4	3	1	75%
16	4	4	0	100%
17	4	4	0	100%
18	4	4	0	100%
19	4	4	0	100%
20	4	4	0	100%
21	4	4	0	100%
22	4	4	0	100%
23	4	4	0	100%
24	4	3	1	75%
25	4	4	0	100%
26	4	3	1	75%
27	4	2	2	50%
28	4	3	1	75%
29	4	3	1	75%
30	4	4	0	100%

31	4	4	0	100%
32	4	4	0	100%
33	4	4	0	100%
34	4	3	1	75%
35	4	4	0	100%
36	4	3	1	75%
37	4	4	0	100%
38	4	3	1	75%
39	4	4	0	100%
40	4	4	0	100%
		Average Recognition Rate		**88.33 %**

Table 6.3 Recognition Performance of the proposed method on ORL face database

6.6 Effect of increased number of epochs

The amount of training also influences the performance of the network. The following table shows the effect of more training time. Though trained for more time, it did not show much improvement in the recognition rate.

Number of Epochs	Elapsed Time(Hr)	Recognition Rate (%)
2000	0.75	88.33
5000	1.805833333	83.33333333
7000	3.6475	83.33333333
10000	5.875277778	85

Table 6.4 Results for other epochs

6.7 Mode of Presenting the Input Patterns

Input patterns can either be presented in the same order during every epoch or the patterns can be shuffled and presented to the network. All the previous results correspond to presenting the shuffled patterns to the network. The effect of un-shuffling the patterns is shown in Table 6.5. The results are not favorable as compared to the shuffled patterns.

Number of Epochs	Elapsed Time(Hr)	Recognition Rate (%)
2000	0.805833333	76.66667
10000	5.875277778	74.16667
15000	7.844722222	76.66667

Table 6.5 Results for un-shuffled patterns

6.8 False Rejection Rate (FRR) and False Acceptance Rate (FAR)

It is necessary to measure the accuracy in order to evaluate the FRS. When performing recognition for verification or authentication purposes, one typically attempts to obtain a score above a fixed threshold. If the score is above or equal to the threshold, the person is recognized, otherwise the person is not recognized.

Accuracy for verification applications is often characterized in terms of two probabilities at a given threshold:

1. False Acceptance Rate (FAR): The chance that an imposter will be erroneously recognized (obtain a matching score equal to or higher than the threshold).

2. False Rejection Rate (FRR): The chance that an authorized person will not obtain a score equal to or above the threshold.

Both the FAR and FRR are functions of threshold. The value where the two probabilities are the same is the Equal Error Rate (EER). The EER is a useful technology descriptor in that it describes performance with a single number. For example, if the EER is 1%, that means 1% of the right people are rejected and 1% of the wrong people are accepted above a certain threshold in a verification task. However, typical real-world implementations do not operate in this equal probability of error regime. Usually, the FAR must be very low but at the sacrifice of a much higher FRR.

To determine the FAR of the proposed system, new images shown in Figure 6-14 are used. These images are from FERET face database. The results (Table 6.6) show the potentiality of the proposed system and it has rejected all the new images and FAR is 0% up to the thresholding limit of 0.5. Also, the table shows FRR corresponding to the test images (120 test images) of the ORL database.

So, the best values for FRR and FAR of the proposed system are 11.7% and 0% respectively. And these optimal values occur at threshold value of 0.5.

Figure 6-15 shows the plot of FAR vs FRR. According to the graph, the intersecting point corresponds to the EER, which is around 0.63. However, as explained earlier, the experimental result is about 0.5 in order to achieve good rejection rate and low acceptance rate.

Figure 6-14: Images from FERET database to measure the FRR of the proposed system

Thresh hold	FRR(%)	FAR(%)
0	100	0
0.1	20	0
0.2	16.66667	0
0.3	13.33333	0
0.4	12.5	0
0.5	11.66667	0
0.6	12.5	7.692308
0.7	15.83333	38.46154
0.8	15.83333	53.84615
0.9	19.16667	61.53846

Table 6.6 FAR and FRR values at different threshold values

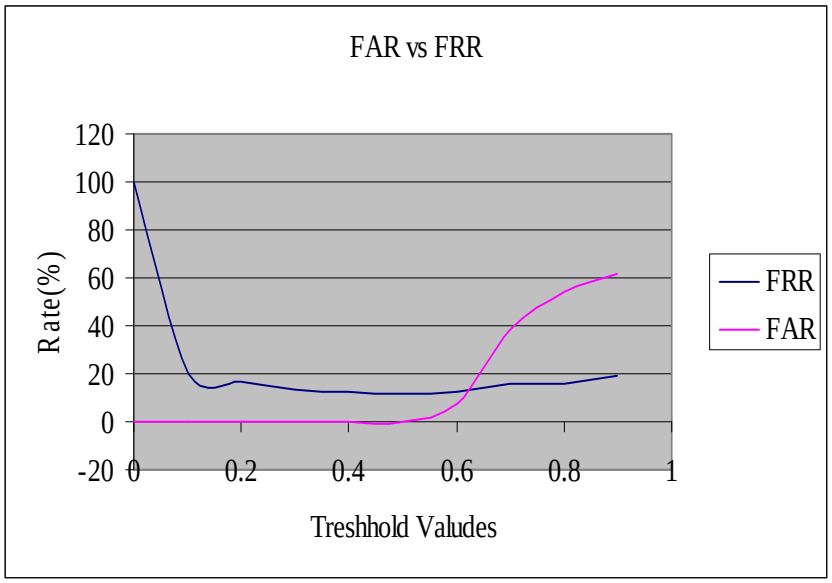


Figure 6-15 FAR vs. FRR

CHAPTER 7 CONCLUSIONS

One of the big advantages with FR is that it is a very natural procedure, and that it is a non-intrusive biometric method that can be done at some distance. FR is a challenging problem and there is still a lot of work that needs to be done in this area. Over the past ten years, FR has received substantial attention from researchers in biometrics, pattern recognition, computer vision, and cognitive psychology communities. This common interest in FR technology among researchers working in diverse fields is motivated both by remarkable ability of humans to recognize people and by the increased attention being devoted to security applications. Applications of FR can be found in security, tracking, multimedia and entertainment domains.

ANN and especially MLP is a promise for the future in pattern recognition. They can recognize patterns within large datasets and then generalize those patterns into recommended courses of action. Designing of neural networks does require a skill. This skill involves a strategy to acquire the necessary data to train the network. It also involves selection of appropriate learning rules, transfer functions, data preprocessing methods and mainly how to connect the neurons within the network.

In this work, 2 FRS are developed. The first model uses Principal Component Analysis (PCA) for feature extraction from the face images and ANN for the classification purpose. In the second model, combination of Gabor Filter (GF) and PCA are used for feature extraction and ANN for the classification.

In the first approach, the face images are projected into subspace called eigenspace, consisting of the eigenvectors from the covariance matrix of all the face images. Since the eigenvectors have a face-like appearance they are called eigenfaces. Eigenvectors with the highest associated eigenvalues represent the highest modes of variation in the dataset of images, and the eigenvectors with the lowest eigenvalues represent the lowest modes of variation. Three different eigenvector lengths (100,150,200 and 300) are investigated and 200 is selected as it provides optimal reconstruction MSE. Then all the images are projected to the eigenspace to transform the image into a representation of a lower dimension which aims to hold the most important features of the face.

In the second approach, the FRS derives a Gabor feature vector based upon a set of downsampled Gabor wavelet representations of face images by incorporating different orientation and scale local features. The Gabor transformed face images exhibit strong characteristics of spatial locality, scale and orientation selectivity, similar to those displayed by the Gabor wavelets. Such characteristics produce salient local features, such as the features in the neighborhood of the eyes, the nose and the mouth that are most suitable for FR. Since the outputs of the Gabor transform consist of different local, scale, and orientation features, we concatenate all these features in order to derive a feature vector. As a result, the feature vector consists of both the real and the imaginary part of the Gabor transform and its length is 655360. Without loss of generality, each output is assumed to be a column vector, which can be constructed by concatenating the rows (or columns) of the output. Before the concatenation, each output is first down sampled by a factor $\rho=64$ to reduce the dimensionality of the original vector space to 10240. PCA operates then on the Gabor feature vector, to reduce its dimensionality. Three different eigenvector lengths (100,150, 200 and 300) are investigated and 200 is selected as it provides optimal reconstruction MSE.

The feature vectors are classified into training, validation and testing sets and stored on files. The classification is based on their intra-subject variation which can be measured using the standard deviation of each feature vector.

The optimum values for the two important parameters (Learning rate and momentum) that control the dynamics of the ANN are determined experimentally and the following points are observed

- While, in general, a smaller η results in slower convergence, it can locate "deeper" local minima in the error surface than a larger η. This finding is intuitively satisfying, since a smaller η implies that the search for a minimum should cover more of the error surface than would be the case for a larger η.
- For $\eta \rightarrow 0$, the use of $\alpha \rightarrow 1$ produces increasing speed of convergence. On the other hand, for $\eta \rightarrow 1$, the use of $\alpha \rightarrow 0$ is required to ensure learning stability.

- The use of the constant $\eta = \{0.5, 0.9\}$ and $\alpha = 0.9$ cause oscillations in the mean squared error during learning and a higher value for the final mean-squared error at convergence, both of which are undesirable effects.

The ANN designed with optimum and necessary parameter is trained with training feature vectors and the validation feature vectors are used to monitor the learning of the network. After successful and proper training, the performance of the network toward its pattern recognition was measured using testing feature vectors and other images. We have also observed the effect of the order of presentation of training feature vectors on the generalization capability of ANN.

Experimentation is carried out on FRS by using Olivetti Research Laboratory (ORL) datasets, the images of which vary in illumination, expression, pose and scale. The result shows the feasibility of the methodology followed in this thesis work. Model 1 achieves a recognition rate of 76.6% whereas model 2 FRS achieves 88.3% of correct classification and performed very efficiently when subjected to new unseen images with a false rejection rate of 0% during testing. The high recognition rate of model 2 shows the efficiency of GF in feature extraction.

CHAPTER 8 RECOMMENDATIONS

There is much space for the improvement of the system:

- Adaptation of the learning rate parameter during training may yield better classification results.
- Can investigate the effect of using more than one hidden layer.
- The algorithm can be improved in order to recognize more complicated images; colored or with different backgrounds.
- The combination of the software with hardware devices, like cameras.
- Creating a graphical user interface makes it more attractive and easier to use
- In order to further speed up the algorithm, number of Gabor filters could be decreased with an acceptable level of performance reduction in recognition.
- It will be an advance if it can provide a graphical output

Appendix A

The ORL Dataset

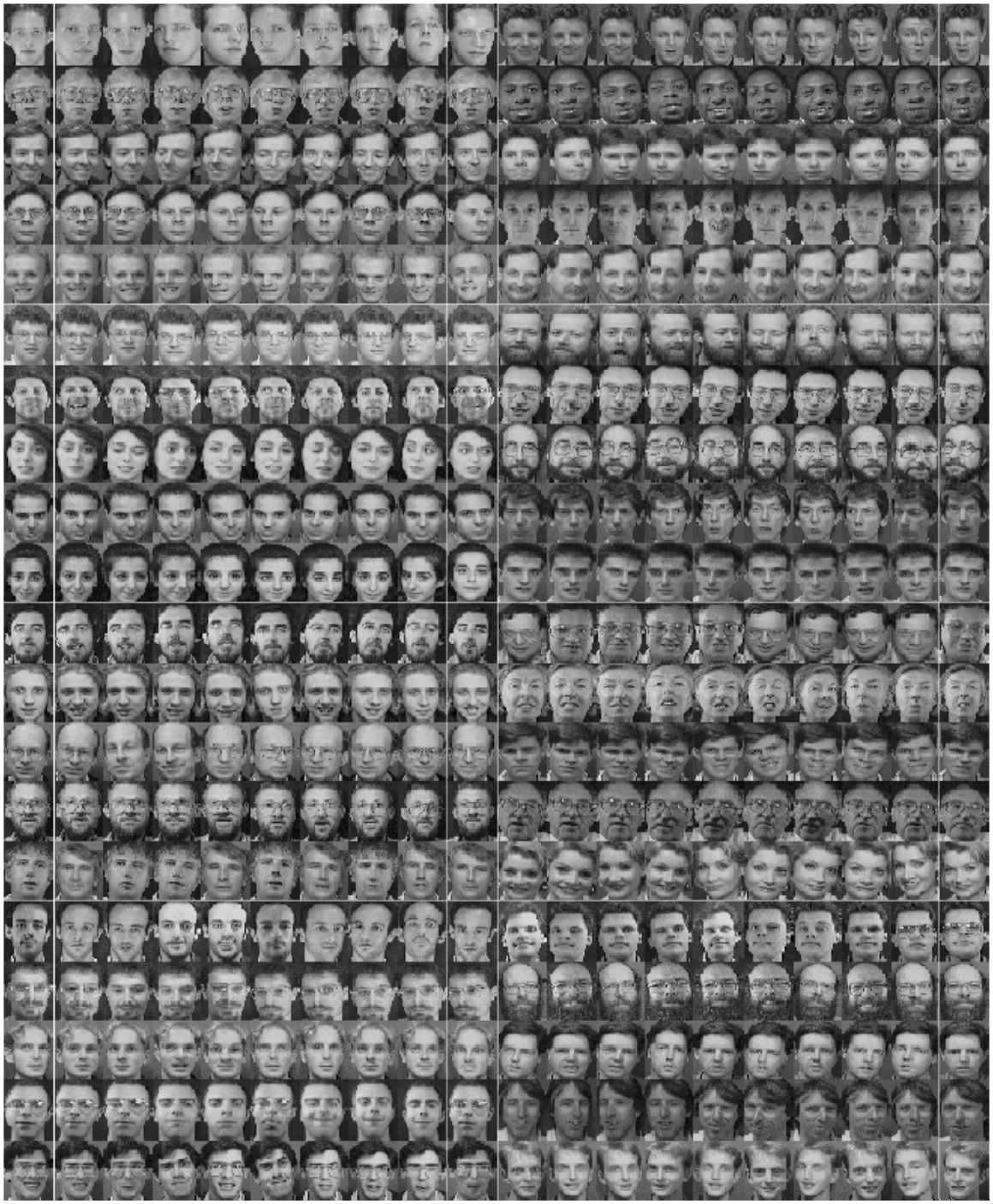


Figure A-1: All Images of The ORL face database

Appendix B

The LMS Leaning Rule

Before developing the complete network learning algorithm called the Generalized Delta Rule (GDR), we first develop the LMS learning rule for a single PE [7, 9].

Suppose we have a set of input vectors, $\{\mathbf{x_1},\mathbf{x_2},\ldots,\mathbf{x_L}\}$, each having its own, perhaps unique, correct or desired output value, d_k, k=1,2,…,L. The problem of finding a single optimum, or best weight vector, **w***, that can successfully associate each input vector with its desired output value. Two methods will be shown to determine **w*** [7, 9]:

i) In analytic or explicit form and

ii) In adaptive iteration form

Both methods use the LMS rule. The second method or procedure, which employs the computation of the weight vector explicitly, is termed as the least mean square (LMS) learning rule to be developed subsequently and the process of finding the weight vector is referred to as training the PE, ALC or neuron [7, 9].

Let us relate the problem a little differently before applying the methods: given examples, $(\mathbf{x_1},d_1)$, $(\mathbf{x_2},d_2)$, …, $(\mathbf{x_L},d_L)$, of some processing function that associates input vectors, $\mathbf{x_k}$, with (or maps to) the desired output values, d_k, what is the best weight vector, **w***, for a PE or an ALC that performs this mapping? [7, 9]

(i) Analytical or Explicit Determination of w*

This method is based on the minimization of the error, the difference between the desired output and the actual output, for each input vector applied to the PE or ALC. The approach selected here is to minimize the mean squared error for the set of input vectors. If the actual output is y_k for the k^{th} input vector $\mathbf{x_k}$, then the corresponding error term is $\varepsilon k = d_k - y_k$.

The mean squared error, or expectation value of the error, is defined by [7]

$$E(\varepsilon_k^{\,2}) = \frac{1}{L}\sum_{k=1}^{L}\varepsilon_k^{\,2} \qquad \text{B.1}$$

Where L is the number of input vectors in the training set,

Using Equation, we can expand the mean squared error as follows:

$$E(\varepsilon_k^2) = E[(d_k - \mathbf{w}^T\mathbf{x}_k)^2] \quad \text{B.2}$$

$$= E(d_k^2) + \mathbf{w}^T E[\mathbf{x}_k\mathbf{x}_k^T]\mathbf{w} - 2E[d_k\mathbf{x}_k^T]\mathbf{w} \quad \text{B.3}$$

In going from Eq. B.2 to Eq. B.3, we have made the assumption that the training set is statistically stationary, meaning that any expectation values vary slowly with respect to time. This assumption allows us to factor out the weight vectors from the expectation value terms in Equation [7, 9].

Now define a matrix $\mathbf{R} = E[\mathbf{x}_k\mathbf{x}_k^T]$, called the input correlation matrix, and a vector $\mathbf{p} = E(d_k\mathbf{x}_k]$ and make the identification $\zeta = E(\varepsilon_k^2)$. Using these definitions, we can rewrite Equations as,

$$\zeta = E[d_k^2] + \mathbf{w}^T\mathbf{R}\mathbf{w} - 2\mathbf{p}^T\mathbf{w} \quad \text{B.4}$$

This equation shows ζ as an explicit function of the weight vector, **w**. In other words, $\zeta = \zeta(\mathbf{w})$.

To find the weight vector corresponding to the minimum mean squared error, we differentiate Equation, evaluate the result at **w***, and set the result equal to zero.

$$\frac{\partial\zeta(\mathbf{w})}{\partial\mathbf{w}} = 2R\mathbf{w}^* - 2\mathbf{p} = 0 \quad \text{B.5}$$

$$R\mathbf{w}^* = \mathbf{p} \quad \text{B.6}$$

$$\mathbf{w}^* = \frac{\mathbf{p}}{R} = R^{-1}\mathbf{p} \quad \text{B.7}$$

Notice that, although ζ is a scalar, $\frac{\partial\zeta(\mathbf{w})}{\partial\mathbf{w}}$ is a vector. Eq. B.5 is an expression of the gradient of ζ, $\nabla\xi$ which is the vector.

$$\nabla\zeta = \left[\frac{\partial\zeta}{\partial w_1}, \frac{\partial\zeta}{\partial w_2}, ..., \frac{\partial\zeta}{\partial w_n}\right]^t \quad \text{B.8}$$

All that we have done by the procedure is to show that we can find a point where the slope of the function $\zeta(\mathbf{w})$, is zero. In general, that point may be a minimum or a maximum point. This result is general and is obtained regardless of the dimension of the

weight vector. In the case of two dimensions, the graph of $\zeta(\mathbf{w})$ is paraboloid and it must be a concave upward surface since all combinations of weights must result in nonnegative value for the mean squared error, ζ. For dimensions higher than two the paraboloid is known as a hyperparaboloid.

Suppose we have an ALC with two inputs and various other quantities defined as follows:

$$R = \begin{bmatrix} 31 \\ 14 \end{bmatrix} \; ; \quad \mathbf{p} = \begin{bmatrix} 4 \\ 5 \end{bmatrix} \; ; (d^2_k)=10$$

Rather than inverting R, we use Eq. (B.10) to find the optimum weight vector:

$$\begin{bmatrix} 3 & 1 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} w_1^* \\ w_2^* \end{bmatrix} = \begin{bmatrix} 4 \\ 5 \end{bmatrix}$$

This equation results in two equations for w^* and w_2^*:

$$3w_1^* + w_2^* = 4$$
$$w^* + 4w_2^* = 5$$

The solution is $w^* = (1, 1)^t$. The graph of ζ as a function of the two weights is shown in Figure B-1. From this figure it can bee seen that for an ALC with only two weights, the error surface is a paraboloid. The weights that minimize the error occur at the bottom of the paraboloidal surface [7].

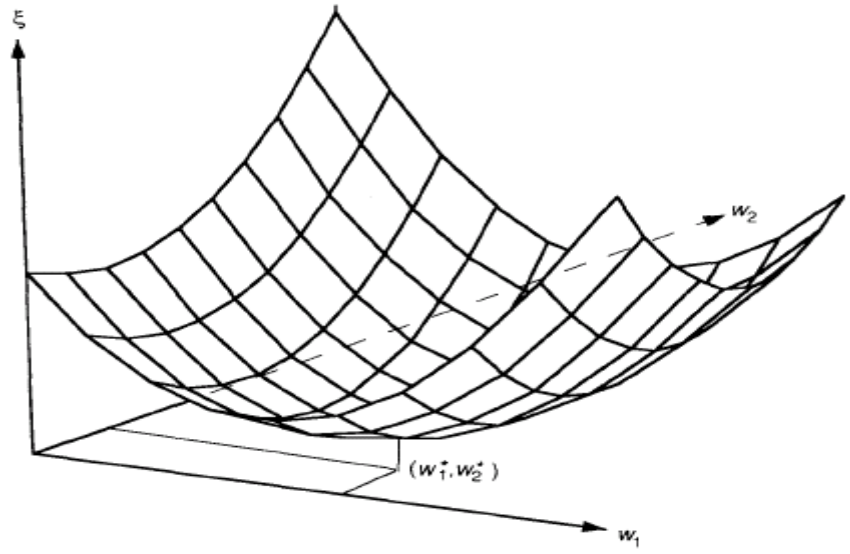


Figure B-1: The error surface for an ALC with two weights [7].

(ii) The Adaptive or Iterative Procedure of the Determination of w*

This method of computing the optimum weight **w*** allows us to avoid the often-difficult calculations necessary to determine the weights manually and is the basis by which most neural network learning algorithms are designed. It employs the method of steepest descent The previous method used to determine **w*** is rather difficult in general. Not only does the matrix manipulation get cumbersome for large dimensions, but also each component of **R** and **p** is itself an expectation value. Thus, explicit calculations of **R** and **p** require knowledge of the statistics of input signals [7]. A better approach would be to let the PE or ALC find the optimum weights itself by having it search over the weight surface to find the minimum. A purely random search might not be productive or efficient, so we shall add some intelligence to the procedure [7].

Begin by assigning arbitrary values to the weights. From that point on the weight surface, determine the direction of the steepest slope in the downward direction. Change the weights slightly so that the new weight vector lies farther down the surface. Repeat the process until the minimum has been reached. This procedure is illustrated in figure B-2

Implicit in this method is the assumption that we know what the weight surface looks like in advance. We do not know the weight surface, however, and we will shortly see the solution to this problem leads to our learning algorithm of the PE [7, 9].

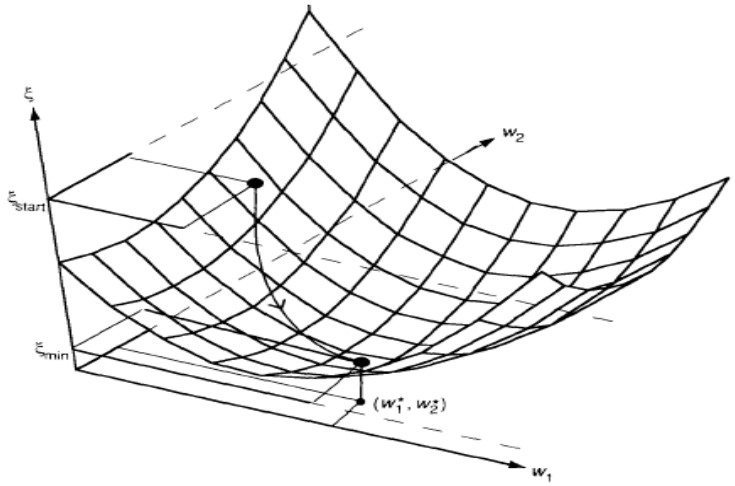


Figure B-2: Visualization of the steepest-descent method [7].

We can use this diagram in figure B-2 to visualize the steepest-descent method. An initial selection for the weight vector results in an error, ζ start. The steepest-descent method consists of sliding this point down the surface toward the bottom, always moving in the direction of the steepest downward slope [7].

Typically, the weight vector does not initially move directly toward the minimum point. The cross-section of the paraboloidal weight surface is usually elliptical, so the negative gradient may not point directly at the minimum point, at least initially. The situation is illustrated more clearly in the contour plot of the weight surface in figure B-3.

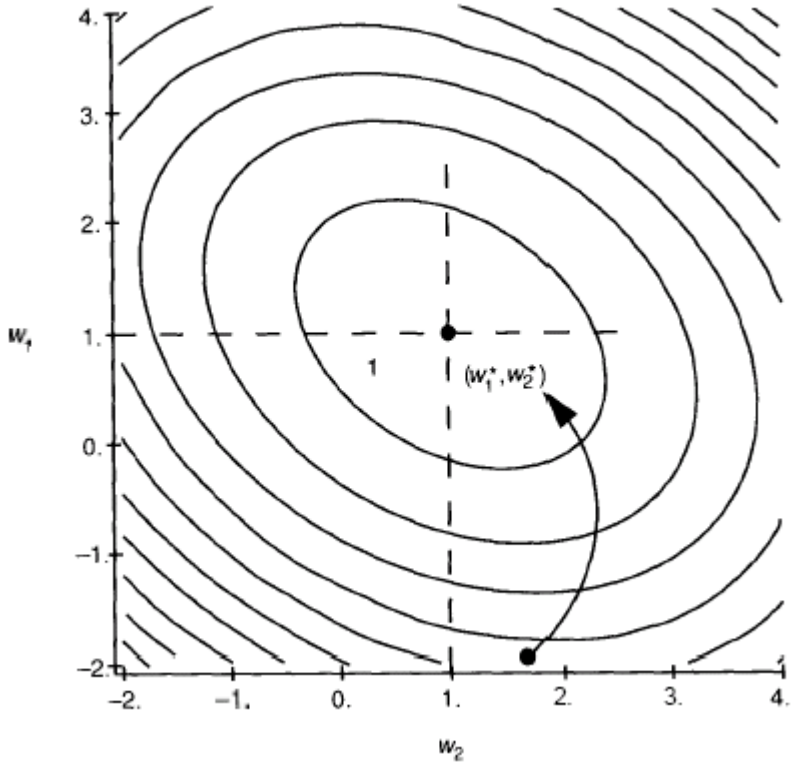


Figure B-3: Contour plot of the weight surface [7].

In the contour plot of the weight surface of figure B-2, the direction of steepest descent is perpendicular to the contour lines at each point, and this direction does not always point to the minimum point [7]. Refer to figure B-3.

Because the weight vector is variable in this procedure, we write it as an explicit function of time step, t. The initial weight vector is denoted **w**(0), and the weight vector at time step t is **w**(t). At each step, the next weight vector is calculated according to [7, 9]

$$\mathbf{w}(t+1) = \mathbf{w}(t) + \Delta\mathbf{w}(t) \qquad \text{B.9}$$

where $\Delta\mathbf{w}(t)$ is the change in **w** at the t^{th} time step.

We are looking for direction of the steepest descent at each point on the surface, so we need to calculate the gradient of the surface (which gives the direction of the steepest upward slope). The negative of the gradient is in the direction of the steepest descent. To get the magnitude of the change, multiply the gradient by suitable constant, η. This procedure results in the following expression:

$$\mathbf{w}(t+1) = \mathbf{w}(t) - \eta \nabla\xi(\mathbf{w}(t)) \qquad \text{B.10}$$

All that is necessary to complete the discussion is to determine the value of $\nabla\xi(\mathbf{w}(t))$ at each successive iteration step. The value of $\nabla\xi(\mathbf{w}(t))$ was determined analytically previously. Equation B.5 or B.8 could be used here to determine $\nabla\xi(\mathbf{w}(t))$ but, we would have the same problem that we had with the analytical determination of **w***; we would need to know both **R** and **p** in advance. This knowledge is equivalent to knowing what the weight surface looks like in advance. To circumvent this difficulty, we use an approximation for the gradient that can be determined from information that is known explicitly at each iteration. For each step in the iteration process, we perform the following [7, 9]:

i) Apply an input vector, $\mathbf{x}_k$, to the ALC inputs.

ii) Determine the value of the error squared, $\varepsilon_k^2(t)$ using the current value of the weight vector

$$\varepsilon_k^2(t) = (d_k - \mathbf{w}^T(t)\mathbf{x}_k)^2 \qquad \text{B.11}$$

iii) Calculate an approximation to $\nabla\xi(t)$, by using $\varepsilon_k^2(t)$ as an approximation for $E[\varepsilon_k^2(t)]$:

$$\nabla\varepsilon_k^{\,2}(t) \approx \nabla E[\varepsilon_k^{\,2}] \qquad \text{B.12}$$

$$\nabla\varepsilon_k^{\,2}(t) = -2\varepsilon_k(t)\mathbf{x_k} \qquad \text{B.13}$$

Where, we have used Equation B.8 to calculate the gradient explicitly.

iv) Update the weight vector according to Equation B.10 using Equation B.13 as the approximation for the gradient

$$\mathbf{w}(t+1) = \mathbf{w}(t) + 2\eta\, \varepsilon_k(t)\mathbf{x}_k \qquad \text{B.14}$$

v) Repeat steps (i) through (iv) with the next input vector, until the error has been reduced to an acceptable value.

Eq. B.14 is an expression of the LMS algorithm. Changes in the weight vector must be kept relatively small on each iteration. If changes are too large, the weight vector could wander about the surface, never finding the minimum, or finding it only by accident rather than as a result of a steady convergence towards it [7].

The parameter η, which is used to prevent this aimless search is called learning rate which has a significant effect on training and determines the stability and speed of convergence of the weight vector towards the minimum-error value. If the statistics of the input signal are known, it is possible to show that its value is restricted to the range [7]

$$\frac{1}{\lambda_{max}} > \eta > 0$$

Where λ_{max} is the largest Eigen value of the input correlation matrix **R**, which assures both the stability and convergence of the weight vector. Since there is no a general guideline as to how choose an appropriate value of η in the absence of the statistics of the input data, experience appears to be the best teacher for its appropriate value [7, 9].

The Generalized Delta Rule

In this section the formal mathematical description of BPN operation is presented with a detailed derivation of the generalized delta rule which is the learning algorithm of the network. The generalized delta rule is a generalization of the LMS rule applied to two or more dimensional problems. Figure B-4 serves as reference for most of the discussions under this topic.

The BPN is a feed forward network that is fully interconnected by layers [6,7]. There are no feedback connections that bypass one layer to go directly to later layer. Although only three layers are used in the discussion, more than one hidden layer is permissible.

A neural network is called a mapping network if it is able to compute some functional relationship between its input and output. For a simple mapping or function we do not need a neural network; however, we might want to perform a complicated mapping where we do not know how to describe the functional relationship in advance, but we do know of examples of the correct mapping. In this situation, the power of a neural network to discover its own algorithms is extremely useful [7].

Suppose we have a set of **P** vector pairs, $(\mathbf{x_1}, \mathbf{y_1}), (\mathbf{x_2}, \mathbf{y_2}), \ldots, (\mathbf{x_p}, \mathbf{y_p})$, which are examples of a functional mapping $\mathbf{y} = \Phi(\mathbf{x})$: $\mathbf{x} \varepsilon \mathbf{e}\ \mathbf{R}^N$, $\mathbf{y} \varepsilon \mathbf{R}^M$. We want to train the network so that it will learn an approximation $\mathbf{o} = \mathbf{y'} = \Phi'(\mathbf{x})$. We shall derive a method of doing this training that usually works provided the training-vector pairs have been chosen properly and there are a sufficient number of them [7].

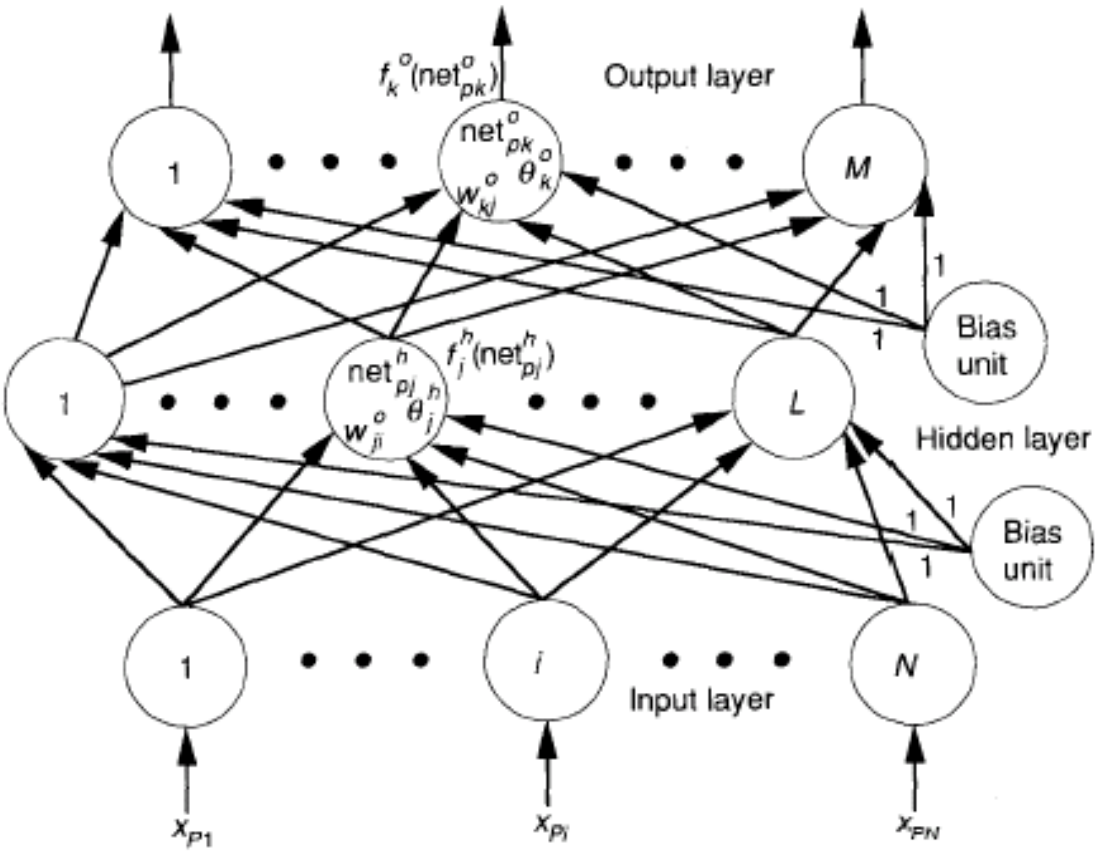


Figure B-4: The three-layer BPN architecture [7].

Figure B-4 shows the three-layer BPN architecture follows closely the general network description. The bias weights, θ^h_j and θ^o_k, and the bias units are optional. The bias units provide a fictitious input value of 1 on a connection to the bias weight. We can then treat the bias weight (or simply, bias) like any other weight: It contributes to the net-input value to the unit, and it participates in the learning process like any other weight.

It should be noted that learning in a neural network means finding an appropriate set of weights. The learning technique that we describe here resembles the problem of finding the equation of a line that best fits a number of known points. For a line-fitting problem, we would probably use a least squares approximation. However, since the relationship we are trying to map is likely to be non-linear, as well as multidimensional, we employ an iterative version of the simple least-square method, called a steepest-descent technique [7, 9].

To begin, let's review the equations for information processing in the three-layer network in Figure B-4. An input vector, $\mathbf{x_p} = (x_{p1},x_{p2},\ldots,x_{pN})^T$, is applied to the input layer of the network. The input units distribute the values to the hidden-layer units. The net input to the j^{th} hidden unit is [7]

$$net^h_{pj} = \sum_{i=1}^{N} w^h_{ji} x_{pi} + \theta^h_j \qquad \text{B.15}$$

Where w^h_{ji} is the weight on the connection from the i^{th} input unit and θ^h_j is the bias term. The "h" subscript refers to the quantities on the hidden layer. Assume that the activation of this node is equal to the net input; then, the output of this node is

$$i_{pj} = f^h_j(net^h_{pj}) \qquad \text{B.16}$$

The equations for the output nodes are

$$net^o_{pk} = \sum_{i=1}^{L} w^o_{kj} i_{pj} + \theta^o_k \qquad \text{B.17}$$

$$o_{pk} = f^o_k(net^o_{pk}) \qquad \text{B.18}$$

Where, the "o" superscript refers to quantities on the outer layer.

The initial set of weight values represents a first guess as to the proper weights for the problem. Unlike some methods, the technique we employ here does not depend on

making a good first guess. There are guidelines for selecting the initial weights, see section 3.5.3. The basic procedure for training the network is embodied in the following description [7].

(i) Apply an input training vector to the network and calculate the corresponding output values.

(ii) Compare the actual outputs with the correct outputs and determine a measure of the error.

(iii) Determine in which direction (+ or -) to change each weight in order to reduce the error.

(iv) Determine the amount by which to change each weight

(v) Apply the corrections to the weights.

(vi) Repeat items (i) through (v) with all training vectors until the error for all vectors in the training set is reduced to an acceptable value.

The iterative weight-change law for network with no hidden units and linear output units, called the LMS rule or delta rule [7]:

$$\mathbf{w}_i(t+1) = \mathbf{w}(t) + 2\eta\varepsilon_k x_{ki} \qquad \text{B.19}$$

where η is a positive constant, x_{ki} is the i^{th} component of the k^{th} training vector, and ε_k is the difference between the actual output, y_k, and the desired or correct value, d_k, that is, $\varepsilon_k = (d_k - y_k)$. Eq. B.19 is just the component form of Eq. B.3

A similar equation results when the network has more than two layers, or when the output functions are non-linear. The results will be derived explicitly in the next section.

Updates of the Output-Layer Weights

In this derivation of the delta rule, since there are multiple units in a layer of a BPN, we shall define the error at a single output unit to be where the subscript "p" refers to the p^{th} training vector, and "k" refers to the k^{th} output unit. In this case, y_{pk} is the desired output value, and o_{pk} is the actual output from the k^{th} unit. The error that is minimized by the GDR is the sum of the squares of the errors for all outputs units [7]:

$$E_p = \frac{1}{2}\sum_{k=1}^{M} \delta^2{}_{pk} \qquad \text{B.20}$$

The factor of ½ in Eq. B.20 is there for convenience in calculating the derivatives later. Since an arbitrary constant will appear in the final result, the presence of this factor does not invalidate the derivation [7].

To determine the direction in which to change the weights, we calculate the negative of the gradient of E_p, ∇E_p with respect to the weights, w_{kj}. Then, we can adjust the values of the weight space [7]. To keep things simple, we consider each component of ∇E_p separately. From Equation B.20 and definition of δ_{pk},

$$E_p = \frac{1}{2}\sum_{k=1}^{M}(y_{pk} - o_{pk})^2 \qquad \text{B.21}$$

and

$$\frac{\partial E_p}{\partial w^o{}_{kj}} = -(y_{pk} - o_{pk})\frac{\partial f^o{}_k}{\partial (net^o{}_{pk})}\frac{\partial (net^o{}_{pk})}{\partial w^o{}_{kj}} \qquad \text{B.22}$$

Where we have used Equation B.18 for the output value, o_{pk}, and the chain rule for partial derivatives.

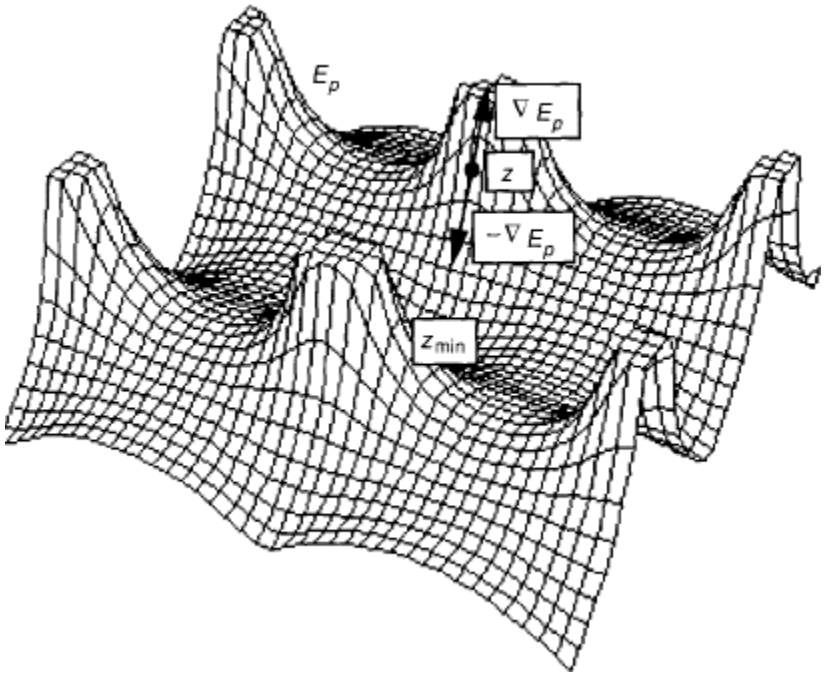


Figure B-5: The hypothetical surface in weight space [7].

Figure B-5 shows the hypothetical surface in weight space hints at the complexity of these surfaces in comparison with the relatively simple hyperparaboloid of the Adaline. The gradient, ∇E_p at point z appears along with the negative of the gradient. Weight changes should occur in the direction of the negative gradient, which is the direction of the *steepest descent* of the surface at the point z. Furthermore, weight changes should be made iteratively until ∇E_p reaches the minimum point z_{min} [7].

For the moment, we shall not try to evaluate the derivative of f^o_k, but instead will write it simply as $f^{o}_{k}{}'(net^o_{pk})$. The last factor in Equation B.22 is

$$\frac{\partial(net^o{}_{pk})}{\partial w^o{}_{kj}} = \left(\frac{\partial}{\partial w^o{}_{kj}} \sum_{j=1}^{L} w^o{}_{kj} i_{pj} + \theta^o{}_k \right) = i_{pj} \qquad \text{B.23}$$

combining Equation B.22 and B.23, we have for the negative gradient

$$-\frac{\partial E_p}{\partial w^o{}_{kj}} = (y_{pk} - o_{pk}) f^o{}_k{}'(net^o{}_{pk}) i_{pj} \qquad \text{B.24}$$

As far as the magnitude of the weight change is concerned, we take it to be proportional to the negative gradient. Thus, the weights on the output layer are updated according to

$$w^o_{kj}(t+1) = w^o_{kj}(t) + \Delta_p w^o_{kj}(t) \qquad \text{B.25}$$

where

$$\Delta_p w^o_{kj}(t) = \eta(y_{pk} - o_{pk}) f^{o}_{k}{}'(net^o_{pk}) i_{pj} \qquad \text{B.26}$$

The factor η is called the learning-rate parameter.

Let's go back to look at the function $f^{o}_{k}{}'$. First, notice the requirement that the function f^o_k be differentiable. This requirement eliminates the possibility of using a linear threshold unit, since the output function for such a unit is not differentiable at the threshold value.
There are two forms of output function that are of interest here [7]:
(i) $f^o_k(net^o_{jk}) = net^o_{jk}$
(ii) $f^o_k(net^o_{jk}) = (1+\exp(-net^o_{jk}))^{-1}$
The first function defines the linear output unit. The later function is called a sigmoid or logistic function; it is illustrated in figure B-6. The choice of output function depends on how you choose to represent the output data. For example, if you want the output units to be binary, you use a sigmoid output function, since the sigmoid is output-limiting and

quasibistable but also differentiable. In other cases, either a linear or a sigmoid output function is appropriate. In the first case, $f^{o}_{k}{}'=1$; in the second case $f^{o}_{k}{}'= f^{o}_{k}(1- f^{o}_{k}) =o_{pk}(1-o_{pk})$ [7].

For these two cases, we have

$$w^{o}_{kj}(t+1) = w^{o}_{kj}(t)+\eta(y_{pk}-o_{pk})i_{pj} \qquad \text{B.27}$$

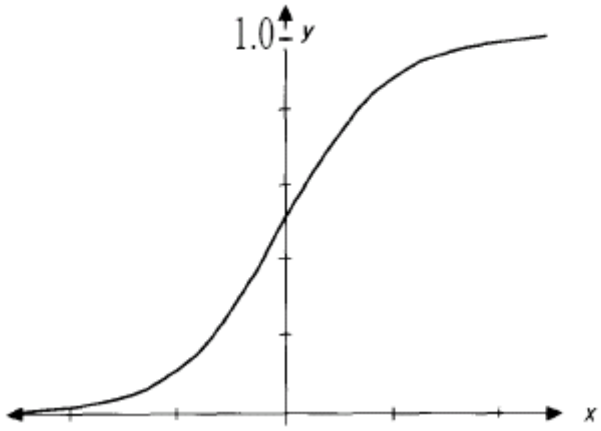


Figure B-6: S-shape characteristic of the sigmoid function [7].

For the linear output, and

$$w^{o}_{kj}(t+1) = w^{o}_{kj}(t) + \eta(y_{pk}-o_{pk})\, o_{pk}\, (1- o_{pk})i_{pj} \qquad \text{B.28}$$

For sigmoidal output.

We want to summarize the weight-update equations by defining a quantity

$$\delta^{o}_{pk} = (y_{pk}-o_{pk})\, f^{o}_{k}{}'(net^{o}_{pk})$$

$$= \delta_{pk}\, f^{o}_{k}{}'(net^{o}_{pk}) \qquad \text{B.29}$$

We can then rewrite the weight-update equation as

$$w^{o}_{kj}(t+1) = w^{o}_{kj}(t)+\eta\delta^{o}_{pk}\, i_{pj} \qquad \text{B.30}$$

Regardless of the functional form of the output function, f^{o}_{k}.

We wish to make a comment regarding the relationship between the gradient-descent method described here and the least-square technique. If we were trying to make the generalized delta rule entirely analogous to a least-square method, we would not actually change any of the weight value until all of the training patterns were had been presented to the network once. We would simply accumulate the changes as each pattern was

processed, sum them, and make one update to the weights. We would then repeat the process until the error was acceptably low [7]. The error that the process minimizes is

$$E = \sum_{p=1}^{P} E_p \qquad \text{B.31}$$

Where, P is the number of patterns in the training set. The procedure is called batch or epoch training. In practice, little advantage is found to this strict adherence to analogy with the least-squares method. Moreover, you must store a large amount of information to use this method. It is recommended, therefore that, you perform weight updates as each training pattern is processed and this procedure is called on-line training [7].

Updates of Hidden Layer Weights

We would like to repeat for the hidden layer the same type of calculation as we did for the output layer. A problem arises when we try to determine a measure of the error of the outputs of the hidden layer units. We know what the actual output is, but we have no way of knowing in advance what correct output should be for these units. Intuitively, the total error, E_p, must somehow be related to the output values on the hidden layer. We can verify our intuition by going back to Eq. B.21

$$\begin{aligned} E &= \frac{1}{2}\sum_k (y_{pk} - o_{pk})^2 \\ &= \frac{1}{2}\sum_k (y_{pk} - f^{o}{}_k(net^{o}{}_{pk}))^2 \\ &= \frac{1}{2}\sum_k (y_{pk} - f^{o}{}_k(\sum_j w_{kj} i_{pj} + \theta^{o}{}_k))^2 \end{aligned}$$

We know that i_{pj} depends on the weights on the hidden layer through Eq. B.15 and B.16. We can exploit this fact to calculate the gradient of E_p with respect to the hidden-layer weight.

$$\begin{aligned} \frac{\partial E_p}{\partial w^{h}{}_{ji}} &= \frac{1}{2}\sum_k \frac{\partial}{\partial w^{h}{}_{ji}} (y_{pk} - o_{pk})^2 \\ &= -\sum_k (y_{pk} - o_{pk}) \frac{\partial o_{pk}}{\partial (net^{o}{}_{pk})} \frac{\partial (net^{o}{}_{pk})}{\partial i_{pi}} \frac{\partial i_{pi}}{\partial (net^{h}{}_{pj})} \frac{\partial (net^{h}{}_{pj})}{\partial w^{h}{}_{ji}} \end{aligned} \qquad \text{B.32}$$

Each of the factors in Equation B.32 can be calculated explicitly from the previous equations. The result is

$$\frac{\partial E_p}{\partial w^h{}_{ji}} = -\sum_k (y_{pk} - o_{pk}) f^{o}{}_{k}{}'(net^o{}_{pk}) w^o{}_{kj} f^{h}{}_{j}{}'(net^h{}_{pj}) x_{pi} \qquad \text{B.33}$$

We update the hidden layer weights in proportion to the negative of Eq. B.33:

$$\Delta_p w^h{}_{ji} = \eta\, f^{h}{}_{j}{}'(net^h{}_{pj}) x_{pi} \sum_k (y_{pk} - o_{pk}) f^{o}{}_{k}{}'(net^o{}_{pk}) w^o{}_{kj} \qquad \text{B.34}$$

where η is once again the learning rate.

We can use the definition of $\delta^o{}_{pk}$ given in the previous section to write

$$\Delta_p w^h{}_{ji} = \eta\, f^{h}{}_{j}{}'(net^h{}_{pj}) x_{pi} \sum_k \delta^o{}_{pk} w^o{}_{kj} \qquad \text{B.35}$$

Notice that every weight update on the hidden layer depends on all the error terms, $\delta^o{}_{pk}$ on the output layer. This result is where the notation of the backpropagation arises. The known errors on the output the propagated back to the hidden layer to determine the appropriate weight changes on that layer. By defining a hidden layer error term

$$\delta^h{}_{pj} = f^{h}{}_{j}{}'(net^h{}_{pj}) \sum \delta^o{}_{pk} w^o{}_{kj} \qquad \text{B.36}$$

We cause the weight update equations to become analogous to those for the output layer:

$w^h{}_{ji}(t+1) = w^h{}_{ji}(t) + \eta\, \delta^h{}_{pj} x_{pi}$

Appendix C

Calculating Eigenfaces

Let a face image $I(x, y)$ be a two-dimensional N by N array of (8-bit) intensity values. Such an image may also be considered as a vector of dimension N^2, so that a typical image of size 128 by 128 becomes a vector of dimension 16,384, or, equivalently, a point in 16,384-dimensional space. An ensemble of images maps to a collection of points in this huge space.

Images of faces, being similar in overall configuration, will not be randomly distributed in this huge image space and thus can be described by a relatively low dimensional subspace. The main idea of the principal component analysis (or Karhunen-Loeve expansion) is to find the vectors which best account for the distribution of face images within the entire image space. These vectors define the subspace of face images called "face space". Each vector is of length N^2, describes an N by N image, and is a linear combination of the original face images. Because these vectors are the eigenvectors of the covariance matrix corresponding to the original face images, and because they are face-like in appearance, they are referred to as "eigenfaces."

As a simple example of this analysis, consider "images" of only three pixels. All possible 1x3 images fill a three-dimensional space. An image of this type is fully specified by three numbers, its coordinates in the 3-D space in Figure C-1(a). If a collection of these images occupy a two-dimensional subspace as in Figure C-1(b), they can be exactly specified by just two numbers, the projections onto the vectors $\mathbf{u}_1$ and $\mathbf{u}_2$ which describe the plane (span the subspace). These vectors are the significant eigenvectors of the covariance matrix of the images. Because they are vectors in the 3-D space, they can also be "displayed" as three-pixel images. A new image which lies near the 2D plane can now be approximately represented by its projection into the plane (or equivalently its projection onto the eigenvectors).

This example is directly analogous to the construction and use of eigenfaces. With real images, the original space has dimension much greater than three, e.g. 16,384-dimensional for 128 by 128 images. The important assumption (supported by [26]) is that

a collection of face images spans some low-dimensional subspace, similar to the plane of points in the example. The eigenvectors (eigenfaces in this case) are 16,384-dimensional, and may be viewed as images.

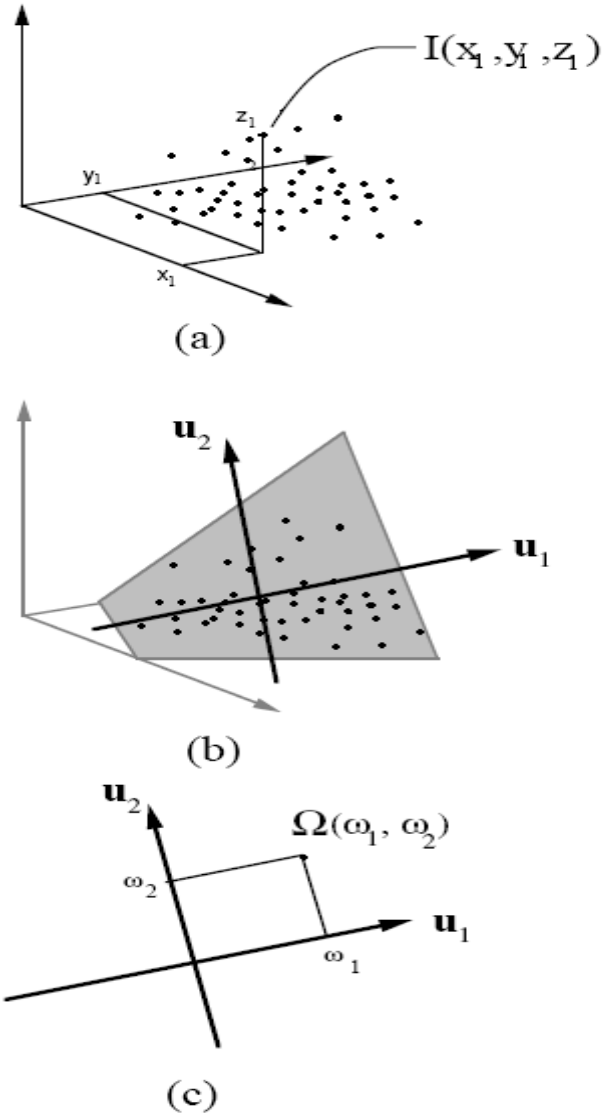


Figure C-1**:** Simple example of PCA. (a) Images with three pixels are described as points in three-space. (b) The subspace defined by a planar collection of these images is spanned by two vectors. One choice for this pair of vectors is the eigenvectors of the covariance matrix of the ensemble, $\mathbf{u}_1$ and $\mathbf{u}_2$. (c) Two coordinates are now sufficient to describe the points, or images: their projections onto the eigenvectors, (ω_1, ω_2) [36].

Let the training set of face images be $\Phi_1, \Phi_2, \Phi_3 \ldots \Phi_M$. The average face of the set is defined by

$$\Psi = \frac{1}{M}\sum_{n=1}^{M}\phi_n. \tag{C.1}$$

Each face differs from the average by the vector

$$\Phi_i = \phi_i - \Psi \tag{C.2}$$

An example training set is shown in Figure C-2(a), with the average face Ψ shown in Figure C-2(b). This set of very large vectors is then subject to principal component analysis, which seeks a set of (*M*-1) orthonormal vectors, $\mathbf{u}_n$, which best describes the distribution of the data. The *k*th vector, $\mathbf{u}_k$, is chosen such that

$$\lambda_k = \frac{1}{M}\sum_{n=1}^{M}(\mathbf{u}_k^t\Phi_n)^2 \tag{C.3}$$

is a maximum, subject to

$$\mathbf{u}_l^t\mathbf{u}_k = \delta_{lk} = \begin{cases} 1, & \text{if } l = k \\ 0, & \text{otherwise} \end{cases} \tag{C.4}$$

for $l < k$, which constrains the vectors to be orthogonal

The vectors $\mathbf{u}_k$ and scalars $\boldsymbol{\lambda_k}$ are the significant *M* eigenvectors and eigenvalues, respectively, of the covariance matrix

$$\begin{aligned} C &= \frac{1}{M}\sum_{n=1}^{M}\Phi_n\Phi_n^t \\ &= AA^t \end{aligned} \tag{C.5}$$

Where the matrix $A = [\Phi 1\ \Phi_2 \ldots \Phi_M]$. The matrix *C*, however, is N^2 by N^2, and determining the N^2 eigenvectors and eigenvalues is an intractable task for typical image sizes. We need a computationally feasible method to find these eigenvectors $\mathbf{u}_i$ of *C*:

$$AA^t\mathbf{u}_i = \lambda_i\mathbf{u}_i \tag{C.6}$$

If the number of data points in the image space is less than the dimension of the space ($M << N_2$), there will be only $M - 1$, rather than N^2, meaningful eigenvectors. (The remaining eigenvectors will have associated eigenvalues of zero.) Fortunately we can solve for the

N^2-dimensional eigenvectors in this case by first solving for the eigenvectors of an M by M matrix — e.g. solving a 16x16 matrix rather than a 16,384 by 16,384 matrix — and then taking appropriate linear combinations of the face images Φ_i. Consider the eigenvectors $\mathbf{v}_i$ of A^tA such that

$$A^t A\mathbf{v}_i = \mu_i \mathbf{v}_i. \tag{C.7}$$

Premultiplying both sides by A, we have [26]

$$AA^t A\mathbf{v}_i = \mu_i A\mathbf{v}_i \tag{C.8}$$

Or

$$AA^t(A\mathbf{v}_i) = \mu_i(A\mathbf{v}_i) \tag{C.9}$$

and comparing with Equation C.6 we see that $A\mathbf{v}_i$ are the eigenvectors of $C = AA_t$. Following this analysis, we construct the M by M matrix $L = A^tA$, where $L_{mn} = \Phi_{tm}\Phi_n$, and find the M eigenvectors, $\mathbf{v}_l$, of L. These vectors determine linear combinations of the M training set face images to form the eigenfaces $\mathbf{u}_l$:

$$\mathbf{u}_l = A\mathbf{v}_i \tag{C.10}$$

With this analysis the calculations are greatly reduced, from the order of the number of pixels in the images (N^2) to the order of the number of images in the training set (M). In practice, the training set of face images will be relatively small ($M<<N^2$), and the calculations become quite manageable. The associated eigenvalues allow us to rank the eigenvectors according to their usefulness in characterizing the variation among the images, and therefore to choose a significant subset to keep.

Figure C-3 shows the top four eigenfaces derived from the input images of Figure C-2. The issue of choosing how many eigenfaces to keep for recognition involves a tradeoff between recognition accuracy and processing time. Each additional eigenface adds to the computation involved in classifying and locating a face. This is not vital for small databases, but as the size of the database increases it becomes relevant [36].

a)

b)

Figure C-2**:** (a) Face images from ORL database (b) The average these faces Ψ

Figure C-3**:** Four of the eigenfaces calculated from the input images of Figure C-2,

REFERENCES

[1]. R. Chellappa, C. Wilson, and S. Sirobey, "Human and machine recognition of faces: A survey," *Proceedings of IEEE*, vol. 83, May 1995.

[2]. F. Galton, "Personal identification and description 1," Nature, pp.173-177, 21 June1888.

[3]. Sir Francis Galton, "Personal identification and description-II", Nature, pp. 201-203, 28 June 1888.

[4]. S. Lawrence, C. Giles, A. Tsoi, and A. Back, "Face Recognition: A Convolutional Neural Network Approach," IEEE *Trans. On Neural Networks*, vol. 8, pp. 98-113, 1997.

[5]. Anil K. Jain, Fellow, IEEE, Robert P.W. Duin, and Jianchang Mao, Senior Member, IEEE, "Statistical Pattern Recognition: A Review," IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 22, No. 1, January 2000.

[6]. GUOQIANG PETER ZHANG, "Neural Networks for Classification: Survey," IEEE Transactions on Systems, Man, and Cybernetics—Part C: Applications and Reviews, Vol. 30, No. 4, November 2000.

[7]. James A. Freeman, David M. Skapura, "Neural Networks Algorithms, Applications, and Programming Techniques," Loral Space Information Systems and Adjunct Faculty, School of Natural and Applied Sciences University of Houston at Clear Lake, Addison-Wesley Publishing Company, 1991.

[8]. Rajesh Parekh, Member, IEEE, Jihoon Yang, Member, IEEE, and Vasant Honavar, Member, IEEE, "Constructive Neural-Network Learning Algorithms for Pattern Classification," IEEE Transactions on Neural Networks, Vol. 11, No. 2, March 2000

[9]. Nura Mustefa, "Finger Print Pattern Recognition using Artificial Neural Network," Addis Ababa University, May, 2005, Addis Ababa.

[10]. Thierry Denoeux, "Pattern Classification," Handbook of Neural Computation IOP Publishing Ltd and Oxford University Press, release 97/1, 2001.

[11]. Klimis Symeonidis, "Hand Gesture Recognition Using Neural Networks," School of Electronic and Electrical Engineering, Germany, August, 2000

[12]. D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar, 'Handbook of Fingerprint Recognition," Springer, New York, 2003.

[13]. Erik Bowman, "Everything You Need to Know About Biometrics," Identix Corporation, January 2000.

[14]. WEICHENG SHEN* AND TIENIU TAN, "Automated Biometrics-based Personal Identification," Identification Technology Division, EER Systems Inc. McLean, VA, Proc. Natl. Acad. Sci. USA, Vol. 96, pp. 11065–11066, September 1999.

[15]. Thomas Fromherz, "Shape from multiple cues for 3-D Enhanced Face Recognition," University of Zurich, 1996

[16]. Johan Blommé, "Evaluation of biometric security systems against artificial fingers," Institutionen för Systemteknik, 2003.

[17]. John D. Woodward, Jr., Christopher Horn, Julius Gatune, and Aryn Thomas, "BiometricsA Look at Facial Recognition," Prepared for the Virginia State Crime Commission, Published 2003 by RAND.

[18]. Keith A, "Rhodes INFORMATION SECURITY Challenges in Using Biometrics," Chief Technologist Applied Research and Methods, United States General Accounting Office, September 2003.

[19]. Bart Leonard, "Biometric Technologies – Evaluating the Solutions," information Security Magazine, March 2000.

[20]. R. Hietmeyer, "Biometric identification promises fast and secure processing of airline passengers," *The Int'l Civil Aviation Organization Journal*, vol. 55, no.9, pp. 10–11, 2000.

[21]. W. Zhao, R. Chellappa, A. Rosenfeld, and P.J. Phillips, "FR: A literature Survey", Technical Report, Univ. of Maryland, 2000.

[22]. J. Huang, "Detection Strategies For FR Using Learning and Evolution", PhD. Thesis, George Mason University, May 1998.

[23]. R. Chellappa, C. L. Wilson and S. Sirohey, "Human and Machine Recognition of Faces: A Survey", Proceedings of the IEEE, Vol. 83, No. 5, May 1995.

[24]. V. Bruce, "Identification of Human Faces", pp. 615-619, Image Processing and Its Applications, Conference Publication No. 465, IEEE, 1999.

[25]. L. Sirovich and M. Kirby, “Low-dimensional procedure for the characterization of human faces,” *J. Opt. Soc. Am. A*, Vol. 4, No. 3, pp. 519-524, March 1987.

[26]. M. A. Sokolov, “Visual motion: algorithms for analysis and application,” Vision and Modeling Group Technical Report #127, MIT Media Lab, February 1990.

[27]. M. Kirby and L. Sirovich, “Application of the Karhunen-Loeve procedure for the characterization of human faces,” *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 12, No. 1, pp. 103-108, Jan. 1990.

[28]. W. Cottrell. P. Munro, and D. Zipser, “Learning internal representations from gray-scale images: An example of extensional programming,” In Proceedings of Annual Conference of the Cognitive Science Society, pages 461-473, Seattle, WA, 1987, Lawrence Erlbaum Associates.

[29]. K. Fleming and G. W. Cottrell, “Categorization of faces using unsupervised feature extraction,” In Proceedings of International Joint Conference on Neural Networks, pages 65-70, San Diego, CA, 1990.

[30]. J. G. Daugman, “Complete discrete 2-D Gabor transforms by neural networks for image analysis and compression,” IEEE Transactions on Acoustics, Speech, and Signal Processing, 36:1169-1179, 1988

[31]. J.-K. Kamarainen, V. Kyrki, and H. K¨alvi¨ainen, “Invariance properties of gabor filter based features - overview and applications,” *IEEE Transactions on Image Processing*, to be published.

[32]. V. Kyrki, J.-K. Kamarainen and H. K¨alvi¨ainen, “Simple Gabor feature space for invariant object recognition,” *Pattern Recognition Letters*, 25(3):311–318, 2004.

[33]. M. Lades, J.C. Vorbr¨uggen, J. Buhmann, J. Lange, C. von der Malsburg, R.P. W¨urtz, andW. Konen, “Distortion invariant object recognition in the dynamic link architecture,” *IEEE Transactions on Computers*, 42(3):300–311, 1993.

[34]. Hyun Jin Park and Hyun Seung Yang, “Invariant object detection based on evidence accumulation and Gabor features,” *Pattern Recognition Letters*, 22:869–882, 2001.

[35]. Ilker Atalay,”FR Using Eigenfaces,” Istanbul Technical University, January 1996

[36]. M. Turk and A. P. Pentland, “FR using eigenfaces,” IEEE Conf. Computer Vision and Pattern Recognition, 1991.

[37]. S.J. McKenna, S. Gong, and J.J. Collins, “Face tracking and pose representation,” In R. B. Fisher and E. Trucco, editors, Proceedings of British Machine Vision Conference, pages 755-764, Edinburgh, 1996. BMVA Press.

[38]. P. J. B. Hancock, V. Bruce, and A. M. Burton, “A comparison of two computer-based face recognition systems with human perceptions of faces,” Vision Research, (Submitted), 1997.

[39]. F.S. Samaria and A.C. Harter, “Parameterization of a stochastic model for human face identification,” In *Proceedings of the 2nd IEEE workshop on Applications of Computer Vision*, Sarasota, Florida, 1994.

[40]. David DeMers and G.W. Cottrell, “Non-linear dimensionality reduction,” In S.J. Hanson, J.D. Cowan, and C. Lee Giles, editors, *Advances in Neural Information Processing Systems 5*, pages 580–587, San Mateo, CA, 1993. Morgan Kaufmann Publishers.

[41]. F.S. Samaria, “*Face Recognition using Hidden Markov Models*,” PhD thesis, Trinity College, University of Cambridge, Cambridge, 1994.

[42]. J. Weng, N. Ahuja, and T.S. Huang “Learning recognition and segmentation of 3-d objects from 2-d images,” In *Proceedings of the International Conference on Computer Vision, ICCV 93*, pages 121–128, 1993.

[43]. Kepenekci, Burcu, “Face Recognition Using Gabor Wavelet Transform,” MSc. Thesis, the Middle East Technical University, September 2001.

[44]. Thomas J., “Biometric Systems”, University of Bologna, January 2002.

[45]. J.-K. Kamarainen, J. Ilonen, and H. K¨alvi¨ainen, “Efficient Computation of Gabor Features”, Lappeenranta University of Technology, 2005